

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA DOPRAVNÍ



AUTOMATIZOVANÁ PROCEDURA MĚŘENÍ POVRCHOVÉ TVRDOSTI
PORÉZNÍCH MATERIÁLŮ

DIPLOMOVÁ PRÁCE

JIŘÍ HOS

PRAHA, 2017



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta dopravní
děkan
Konviktská 20, 110 00 Praha 1

K618Ústav mechaniky a materiálů

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení studenta (včetně titulů):

Bc. Jiří Hos

Kód studijního programu a studijní obor studenta:

N 3710 – IS – Inteligentní dopravní systémy

Název tématu (česky): **Automatizovaná procedura měření povrchové
tvrdosti porézních materiálů**

Název tématu (anglicky): Automated Procedure for Hardness Mapping of the Porous
Materials

Zásady pro vypracování

Při zpracování diplomové práce se řiďte osnovou uvedenou v následujících bodech:

- Cílem diplomové práce je algoritmizace automatické procedury pro mapování povrchové tvrdosti porézních materiálů na základě indentačních zkoušek.
- Vhodné rozmístění indentů bude navrženo na základě zpracování obrazových dat testovaného vzorku, experimentální procedura bude časově optimalizována.
- Aplikace bude vytvořena pomocí vhodného nástroje (softwarový nástroj Matlab, nebo např. programovací jazyk Python).
- Výstupem bude zdrojový kód (G-kód) pro počítačem číslicově řízenou (CNC) experimentální proceduru s ověřenou funkčností pomocí indentoru sestrojeného na ústavu K618.

Rozsah grafických prací: nebyl stanoven

Rozsah průvodní zprávy: minimálně 55 stran textu (včetně obrázků, grafů a tabulek, které jsou součástí průvodní zprávy)

Seznam odborné literatury: John C. Russ, F. Brent Neal, The Image Processing Handbook, Seventh Edition, CRC Press, 2015
William N. Sharpe, Springer Handbook of Experimental Solid Mechanics, 2008th Edition, Springer
Michael J Peterson, CNC Programming: Basics & Tutorial Textbook, 2008

Vedoucí diplomové práce: **Ing. Nela Krčmářová**
Ing. Daniel Kytýř, Ph.D.

Datum zadání diplomové práce: **15. července 2016**
(datum prvního zadání této práce, které musí být nejpozději 10 měsíců před datem prvního předpokládaného odevzdání této práce vyplývajícího ze standardní doby studia)

Datum odevzdání diplomové práce: **30. května 2017**
a) datum prvního předpokládaného odevzdání práce vyplývající ze standardní doby studia a z doporučeného časového plánu studia
b) v případě odkladu odevzdání práce následující datum odevzdání práce vyplývající z doporučeného časového plánu studia

prof. Ing. Ondřej Jiroušek, Ph.D.
vedoucí
Ústavu mechaniky a materiálů



prof. Dr. Ing. Miroslav Svítek, dr. h. c.
děkan fakulty

Potvrzuji převzetí zadání diplomové práce.

Bc. Jiří Hos
jméno a podpis studenta

V Praze dne.....15. července 2016

Prohlášení

Předkládám tímto k posouzení a obhajobě diplomovou práci, zpracovanou v závěru studia na ČVUT v Praze, Fakultě dopravní.

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

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č.121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorských zákonů).

Praha, 29. května 2017

.....

Jiří Hos

Poděkování

Rád bych poděkoval všem, kteří mi pomáhali při vzniku této práce. Jmenovitě se jedná o vedoucí mé práce Ing. Nelu Krčmářovou a Ing. Daniela Kytýře, Ph.D. Dále bych rád poděkoval Ing. Janu Šleichrtovi za cenné poznatky, které usnadnily vytváření této diplomové práce.

V neposlední řadě bych rád poděkoval rodině a přátelům, kteří mi poskytovali podporu v průběhu mého dosavadního studia.

Tato práce byla podpořena grantem Studentské grantové soutěže ČVUT č. SGS15/225/OHK2/3T/16.

AUTOMATIZOVANÁ PROCEDURA MĚŘENÍ POVRCHOVÉ TVRDOSTI PORÉZNÍCH
MATERIÁLŮ

JIŘÍ HOS

ČVUT v Praze, Fakulta dopravní

Praha, 2017

Abstrakt

Zkoušky určování povrchové tvrdosti představují jedny z nejpoužívanějších zkoušek materiálů. Výhodou tohoto typu testu je jeho minimální destruktivnost, nízké náklady na jeho provádění a relativně jednoduché vyhodnocení.

Tato diplomová práce se zabývá vytvořením automatizované procedury měření povrchové tvrdosti porézních materiálů. Tyto poznatky jsou využity pro napsání programu pro určování nejvhodnějších lokací pro indentaci. Vstupy pro vytvořený program jsou fotografie porézního materiálů. Složením těchto snímků je vytvořen pracovní obraz na němž jsou následně prováděny procedury sloužící k identifikaci nejvhodnějších lokací pro indent. Výstupem skriptu je G-kód pro CNC proceduru.

Vytvořený G-kód je použit pro experimentální měření tvrdosti. Vytvořené indenty jsou nafoceny, poté jsou změřeny velikosti jejich úhlopříček, které poslouží jako základ pro výpočet tvrdosti podle Vickerse.

Klíčová slova

Automatizace, měření povrchové tvrdosti, porézní materiály

AUTOMATED PROCEDURE FOR HARDNESS MAPPING OF THE POROUS MATERIALS

JIŘÍ HOS

CTU in Prague, Faculty of Transportation Sciences

Prague, 2017

Abstract

Hardness tests belong among the most used tests of materials. The advantages of this type of test are its minimal destructiveness and its relatively low price.

This diploma thesis is focused on creation of automated procedure for hardness mapping of porous materials. Based on this knowledge program for assessment of the most suitable location for indentation was developed. Inputs to the tool are images of porous material. Final image, that is used for identification of locations for indentation, is composed of this subimages. Output of software tool is G-code for CNC procedure.

The G-code is used for experimental hardness mapping. Indents are optically inspected and size of their diagonal are measured. Finally hardness is computed by application of Vickers hardness equation.

Keywords

Automation, hardness mapping, porous materials

Obsah

1	Úvod	16
1.1	Motivace	16
1.2	Cíle práce	16
1.3	Struktura práce	17
2	Porézní materiály	18
2.1	Druhy porézních materiálů	18
2.2	Struktura porézních materiálů	19
2.2.1	Porozita	19
2.2.2	Specifický povrch	20
2.2.3	Rozložení velikosti pórů	20
2.3	Vlastnosti porézních materiálů	20
2.3.1	Mechanické vlastnosti	20
2.3.2	Tepelné vlastnosti	23
2.3.3	Elektrické vlastnosti	23
2.4	Výroba	24
2.5	Využití porézních materiálů	25
2.5.1	Automobilový průmysl	25
2.5.2	Letecký průmysl	25
2.5.3	Kosmický průmysl	25
2.5.4	Námořní průmysl	26
2.5.5	Železniční průmysl	26
2.5.6	Sportovní vybavení	26
2.5.7	Biomedicínský průmysl	26

3	Povrchová tvrdost	27
3.1	Metody vnikací	29
3.1.1	Brinellova metoda	30
3.1.2	Vickersova metoda	33
3.1.3	Rockwellova metoda	35
3.1.4	Knoopova metoda	37
3.1.5	Metoda universální tvrdosti	39
4	Teorie zpracování obrazových dat	40
4.1	Pořízení obrazu	40
4.2	Chyby obrazu	41
4.3	Segmentace a Thresholding	41
4.3.1	Thresholding	41
4.4	Binární obraz	43
4.5	Základní morfologické operace	43
4.5.1	Otevření	44
4.5.2	Uzavření	44
4.5.3	Dilatace	44
4.5.4	Eroze	45
5	Vývoj nástroje pro zpracování obrazových dat	46
5.1	Grafický výstup	47
5.2	Nastavení zpracování obrazu	48
5.3	Uživatelské menu	48
5.3.1	Calibration	49
5.3.2	Load image	49
5.3.3	Show binary image	50
5.3.4	Execute	51
5.3.5	Add point	53
5.3.6	Delete point	53
5.3.7	Export	53
5.3.8	Exit	53
5.4	Hledání nejkratší cesty	54
5.4.1	Hamiltonovská kružnice	55

5.4.2	Genetický algoritmus	55
5.4.3	Hladový algoritmus	55
5.5	Dopočítání souřadnic	56
5.6	Výstup	57
6	Experiment	58
6.1	Automatizovaný přístroj - instrumentovaný indentor	58
6.1.1	Mechanická část	59
6.1.2	Optická část	60
6.1.3	Enkodér	60
6.2	Modulární jednotka	60
6.2.1	Vlastnosti řídicí jednotky	60
6.3	G-kód	62
6.3.1	Vysvětlení kódu	62
6.4	Materiál	63
6.5	Příprava vzorku	64
6.6	Popis kalibrace	64
6.7	Popis experimentu	64
7	Výsledky	67
7.1	Vyhodnocení experimentu	67
7.2	Hodnocení chyby polohování	68
7.2.1	Indenty ve čtvercové mřížce	69
7.2.2	Indenty v řadě	70
8	Závěr	72
8.1	Zhodnocení softwarového nástroje	72
8.2	Zhodnocení experimentu	72
8.3	Zhodnocení přesnosti polohování	73
	Literatura	74
	Příloha A - Matlab	78
	Příloha B - G-kód	101

Seznam obrázků

2.1	Členění porézních materiálů na základě velikosti pórů	19
2.2	Srovnání porézních materiálů s různými relativními hustotami	21
2.3	Závislost hustoty na Youngově modulu	21
2.4	Závislost hustoty na pevnosti v tlaku	22
2.5	Závislost pevnosti na tuhosti	22
2.6	Tepelná vodivost porézních materiálů	23
3.1	Srovnání škál tvrdosti	28
3.2	Vztah mezi tvrdostí a pevností v tahu	29
3.3	Průběh zkoušky tvrdosti podle Brinella	31
3.4	Průběh zkušebního zatížení během provádění zkoušky podle Brinellovy metody	32
3.5	Průběh zkoušky tvrdosti podle Vickerse	34
3.6	Zařízení pro zkoušku tvrdosti podle Rockwella	36
3.7	Průběh zkoušky tvrdosti podle Rockwella	36
3.8	Průběh zkušebního zatížení během měření tvrdosti podle Rockwellovy metody	37
3.9	Knoopova metoda	38
4.1	Histogram s dvěma mody	42
4.2	Binární obraz	43
4.3	Obrazová data před a po provedení operace Otevření	44
4.4	Obrazová data před a po provedení operace Uzavření	45
5.1	Grafické uživatelské rozhraní	47
5.2	Grafický výstup	48
5.3	Nastavení zpracování obrazu	48
5.4	Obraz s měrkou	49
5.5	Dílčí obraz	50

5.6	Pracovní obraz, červeně jsou znázorněny hranice mezi výřezy z dílčích obrazů . . .	51
5.7	Grafické uživatelské rozhraní - funkce Export	54
6.1	Instrumentovaný indentor	59
6.2	Modulární jednotka	61
6.3	Modulární jednotka - schéma	61
6.4	Vzorová struktura G-kódu	63
6.5	Pracovní obraz	65
6.6	Pracovní obraz s identifikovanými lokacemi pro indentaci	66
6.7	Snímky indentů	66
7.1	Příklad aplikace skriptu pro měření velikosti tvrdosti	67
7.2	Příklad špatně provedené indentace	68
7.3	Indenty ve čtvercové mřížce	69
7.4	Indenty v řadě	70

Seznam tabulek

7.1	Tabulka naměřených a vypočtených hodnot	68
7.2	Měření délky o předpokládané velikosti 0,75 mm	69
7.3	Měření délky o předpokládané velikosti 0,375 mm	70
7.4	Měření délky o předpokládané velikosti 1 mm	71
7.5	Měření délky o předpokládané velikosti 0,66 mm	71
7.6	Měření délky o předpokládané velikosti 0,33 mm	71

Seznam použitých pojmů

Pojem	Vysvětlení
G - kód	programovací jazyk pro CNC obráběcí stroje
Matlab	programové prostředí
Porézní materiál	materiál obsahující velké množství malých pórů
Tvrдост	odpor proti deformaci
Zkouška tvrdosti	zkouška při které je zjišťována velikost tvrdosti
Image processing	zpracování obrazu
Segmentace	členění obrazů do menších celků na základě jejich vlastností
Thresholding	proces sloužící k vytvoření černobílého obrazu
Otevření	odstranění drobných objektů
Uzavření	vyplnění prázdných prostorů
Eroze	logický rozdíl
Dilatace	logický součet
Problém obchodního cestujícího	úloha řešící hledání nejkratší cesty spojující všechny body v oblasti

Seznam použitých veličin

Symbol	Jednotka	Veličina
ϕ	[-]	porozita
V_v	[m ³]	objem pórů
V_s	[m ³]	objem materiálu
ρ_r	[-]	relativní hustota
ρ	[kg · m ³]	hustota pěny
ρ_s	[kg · m ³]	hustota materiálu
λ	[W · m ⁻¹ · K ⁻¹]	tepelná vodivost pěny
λ_s	[W · m ⁻¹ · K ⁻¹]	tepelná vodivost materiálu
R	[Ω · m]	rezistivita pěny
R_s	[Ω _s · m]	rezistivita materiálu
HB	[-]	velikost tvrdosti podle Brinella
HBW	[-]	velikost tvrdosti podle Brinella s hrotem z wolframu
HBS	[-]	velikost tvrdosti podle Brinella s hrotem z oceli
D	[mm]	průměr zkušebního tělesa
d	[mm]	průměr vtisku
HV	[-]	velikost tvrdosti podle Vickerse
F	[N]	zkušební zatížení
HR	[-]	velikost tvrdosti podle Rockwella
h	[mm]	rozdíl výšek mezi různými zatíženími při měření tvrdosti podle Rockwella
HB	[-]	velikost tvrdosti podle Knoopa
HU	[-]	velikost tvrdosti podle universální metody
u	[mm]	velikost úhlopříčky indentu
$chyba$	[%]	chyba měření
l	[mm]	vzdálenost mezi indenty

Kapitola 1

Úvod

Tato diplomová práce se zabývá vytvořením automatizované procedury k měření povrchové tvrdosti porézních materiálů.

1.1 Motivace

Porézní materiály se vyznačují svojí schopností pohlcovat mechanickou energii. Z tohoto důvodu jsou velkou měrou využívány v dopravních aplikacích [1].

V této diplomové práci je řešena problematika automatizace zkoušek tvrdosti tohoto typu materiálu. Zkoušky tvrdosti se používají kvůli své částečné nedestruktivnosti a nízkým nákladům. Z těchto důvodů se jedná o jedny z nejčastěji prováděných zkoušek vlastností materiálů. Vhodným zautomatizováním celého procesu by mohlo dojít k výraznému urychlení celého testu.

1.2 Cíle práce

Primárním cílem této diplomové práce byla algoritmizace automatické procedury pro mapování povrchové tvrdosti porézních materiálů na základě indentačních zkoušek. Tento hlavní cíl měl několik dílčích úkolů. Prvním z těchto dílčích úkolů bylo vytvoření vhodného softwarového nástroje, který by na základě zpracovaných obrazových dat pořízených na testovaném vzorku navrhoval nejvhodnější rozmístění indentů.

Dalším dílčím úkolem byla časová optimalizace vytvořeného softwarového nástroje. Pro splnění těchto dílčích úkolů bylo použito výpočetní prostředí softwarového nástroje Matlab (Matlab R2012a, MathWorks, Inc., Natick, USA). Využit byl i balík procedur pro tento nástroj označovaný jako Image Processing Toolbox. Posledním dílčím úkolem této diplomové práce bylo ověření

funkčnosti výstupního zdrojového kódu, tzv. G-kódu, pro počítačem číslicově řízenou experimentální proceduru pomocí indentoru, který byl sestaven v rámci předchozí vědeckovýzkumné práce ústavu K618. Zmíněný zdrojový kód je automaticky generován jako výstup tohoto softwarového nástroje.

1.3 Struktura práce

V prvních třech kapitolách této diplomové práce - Porézní materiály, Povrchová tvrdost a Teorie zpracování obrazových dat, je čtenář seznámen s teoretickými podklady nezbytnými pro zpracování této práce, tedy z vlastnostmi porézních materiálů, s nejrozšířenějšími metodami pro určování povrchové tvrdosti materiálů a základními přístupy pro vhodnou úpravu obrazových dat, aby z nich mohly být získány požadované informace.

Po té následují dvě kapitoly, které se zabývají vyřešením dílčích úkolů, které byly zadány pro tuto diplomovou práci. V první z těchto kapitol - Vývoj nástroje pro zpracování obrazových dat, je popsán softwarový nástroj, který slouží k identifikaci nejvhodnějších lokací pro provedení indentace. V této kapitole je rovněž popsán výstup tohoto nástroje a přístupy, pomocí kterých byla řešena časová optimalizace tohoto nástroje. V druhé z těchto kapitol, které se zabývají praktickou částí práce, je popsán zdrojový kód pro počítačem řízenou experimentální proceduru a instrumentovaný indenter. Rovněž je v této kapitole popsána příprava vzorku a samotný průběh experimentu, v rámci kterého je ověřována funkčnost navrženého algoritmu.

V kapitole Výsledky je ve stručnosti vyhodnocen celý experiment a jsou zde popsány jeho výsledky. V této kapitole jsou rovněž ve zkratce popsány výsledky kalibrace. Součástí této kapitoly je zhodnocení přesnosti a spolehlivosti navrženého algoritmu.

V poslední části této diplomové práce, v kapitole Závěr jsou shrnuty závěry a poznatky získané během vytváření algoritmu pro automatickou proceduru, která slouží k mapování povrchové tvrdosti porézních materiálů na základě indentačních zkoušek.

Kapitola 2

Porézní materiály

Tato kapitola se zabývá jednak popisem porézních materiálů a jednak popisem jednoho z jejich zástupců - kovovými pěnamí. Porézní materiály lze popsat jako pevné látky obsahující póry. Materiály tohoto typu se hojně vyskytují v přírodě a značnou měrou jsou využívány i v průmyslové praxi [2], kde se využívají především pro svoji schopnost pohlcovat mechanickou energii [3].

Pěnamí se v původním významu slova rozumí disperze plynu v kapalině. V případě, že necháme tuto pěnu ztuhnout, získáme tzv. pevnou pěnu [4].

Pěny a obecně porézní materiály jsou takové materiály, které se vyznačují tím, že obsahují poměrně velké množství malých pórů [5]. Kovové pěny se spolu s ostatními porézními materiály vyznačují nízkou hustotou vysokou tuhostí, nízkou měrnou hmotností, vysokou propustností plynu a vysokou tepelnou vodivostí. Proto se lze vzhledem k těmto vlastnostem v přírodě setkat s materiály s podobnou celulární strukturou, na příklad ve dřevě nebo v kostech [4]. Vlastnosti porézních materiálů je předurčují k využití v lehkých konstrukcích, v aplikacích kdy je třeba absorbovat velké množství energie [6].

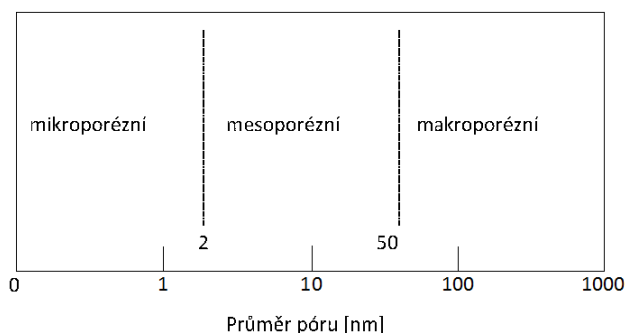
V současné době tvoří základ drtivě většiny kovových pěn hliník nebo nikl, ale pěny lze vyrábět i z jiných prvků jako jsou třeba hořčík, olovo, zinek, měď, bronz, titan, ocel, a nebo zlato [6].

2.1 Druhy porézních materiálů

Porézní materiály lze členit podle různých kritérií. Nejpoužívanějšími kritérii jsou jednak velikosti pórů a jednak základní látka, ze které jsou materiály vyrobeny [7].

Z hlediska velikosti pórů, které se nacházejí uvnitř materiálu, můžeme porézní materiály členit do tří kategorií, viz obrázek 2.1. Těmito kategoriemi jsou materiály mikroporézní, meso-

porézní a makroporézní [2].



Obrázek 2.1: Členění porézních materiálů na základě velikosti pórů [2]

Druhým způsobem, jak členit porézní materiály je na základě látky z které jsou vytvořeny [8]. Porézní materiály tak mohou být rozděleny do následujících kategorií:

- přírodní materiály (dřevo, kosti, porézní horniny, půdy)
- umělé materiály (beton, keramika, kovové pěny, polymerní pěny) [8]

2.2 Struktura porézních materiálů

Porézní materiály se vyznačují svojí typickou topologií buněk, relativní hustotou, velikostí a tvarem buněk a anisotropií. Z těchto parametrů se nejnázve zjišťuje velikost hustoty k jejímuž výpočtu potřebujeme pouze vzorek materiálu, u kterého známe jeho hmotnost a objem. K zjištění ostatních vlastností je třeba využít mikroskopie, ať už optické, a nebo elektronové [6].

Nejdůležitějšími parametry s jejichž pomocí lze popsat vlastnosti porézních materiálů jsou porozita, specifický povrch a rozložení velikostí pórů. Nicméně výše zmíněné parametry nestačí k popisu vlastností porézních materiálů. Velkou roli má i například tvar pórů [8].

2.2.1 Porozita

Vlastností zvanou porozita se rozumí podíl pórů v celkovém objemu materiálu. Velikost této veličiny se obvykle určuje porovnáním hmotnosti suchého vzorku s vzorkem ponořeným do kapaliny o známé hustotě. Nejběžnější hodnoty porozity spadají do intervalu od 0,20 do 0,95. Porozita je popsána následujícím vztahem:

$$\phi = \frac{V_v}{V_s}, \quad (2.1)$$

kde ϕ je porozita, V_v je celkový objem pórů a V_s je celkový objem materiálu [2].

2.2.2 Specifický povrch

Specifický povrch je veličina, která udává celkovou velikost plochy na jednotku hmoty. Tento parametr má význam pro aplikaci porézních materiálů jakožto filtrů. Metodou, která se nejvíce používá pro stanovení specifického povrchu, je adsorpce plynů [2], což je proces během kterého se plynná látka hromadí na povrchu pevné látky vlivem mezipovrchových přitažlivých sil.

2.2.3 Rozložení velikosti pórů

Termínem rozložení velikosti pórů se rozumí četnost výskytu pórů s různou velikostí ve struktuře zkoumaného porézního materiálu [7]. K určení četnosti výskytu pórů se používají různé metody, těmi nejvyužívanějšími jsou rtuťová porozimetrie [9], adsorpce plynu [10], adsorpce vodního páry [11], případně pozorování struktury porézního materiálu.

2.3 Vlastnosti porézních materiálů

Vlastnosti konkrétních porézních materiálů lze nejnázorněji popsat pomocí porovnání vlastností původního a porézního materiálu. Dalším parametrem, pomocí kterého lze popsat vlastnosti porézních materiálů, je relativní hustota ρ_r , viz obrázek 2.2, která je popsána vztahem:

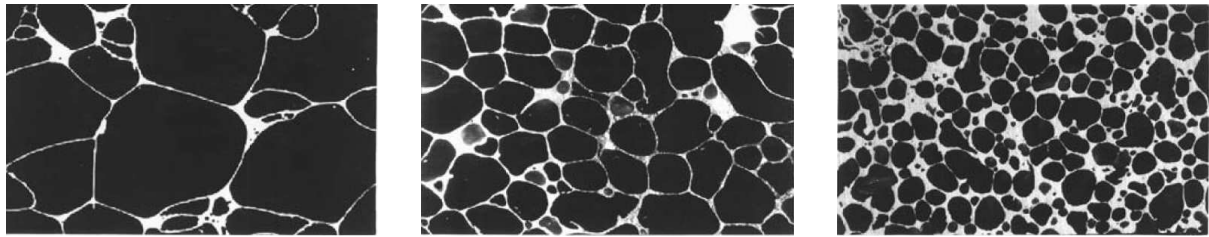
$$\rho_r = \frac{\rho}{\rho_s}, \quad (2.2)$$

kde ρ je hustota pěny a ρ_s je hustota materiálu. Vlastnosti porézních materiálů lze popsat rovněž pomocí vlastností jejich buněk, tedy jestli jsou otevřené, nebo uzavřené. Vlastnosti porézních materiálů jsou dále ovlivňovány jejich strukturou, jednak anisotropií a jednak defekty ve struktuře, jimiž se rozumí různé zkroucení a zlomení buněčné stěny, anebo obecně buňky nepravidelného tvaru [6].

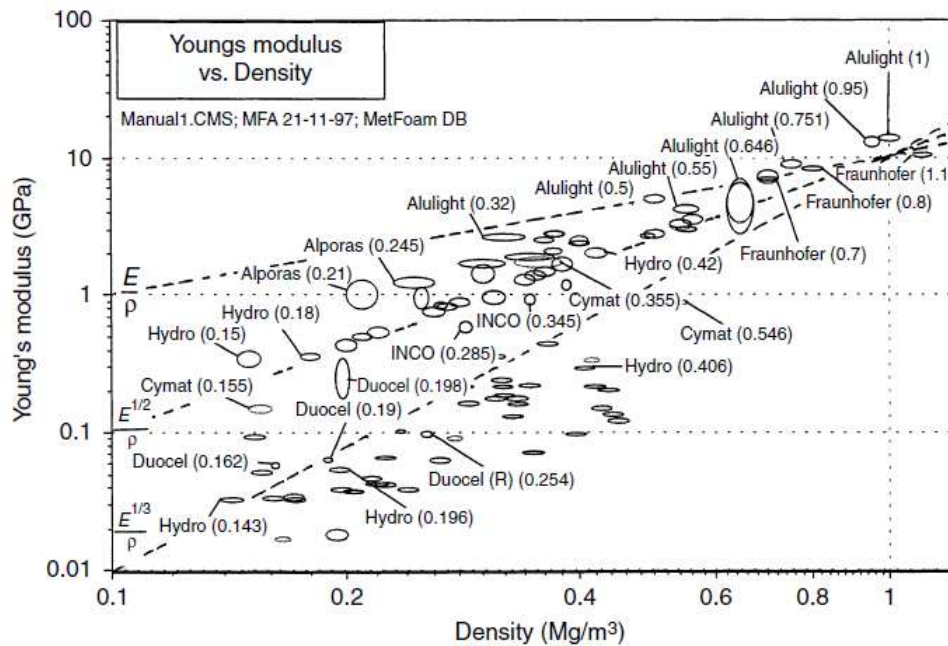
2.3.1 Mechanické vlastnosti

Závislost tuhosti na hustotě

Porézní materiály se vyznačují dobrým poměrem tuhosti a hustoty, viz obrázek 2.3, což je činí vhodným pro použití jako lehké a zároveň tuhé desky. Zároveň se nabízejí jako možnost pro zvýšení vlastní frekvence [6].



Obrázek 2.2: Srovnání porézních materiálů s různými relativními hustotami [6]



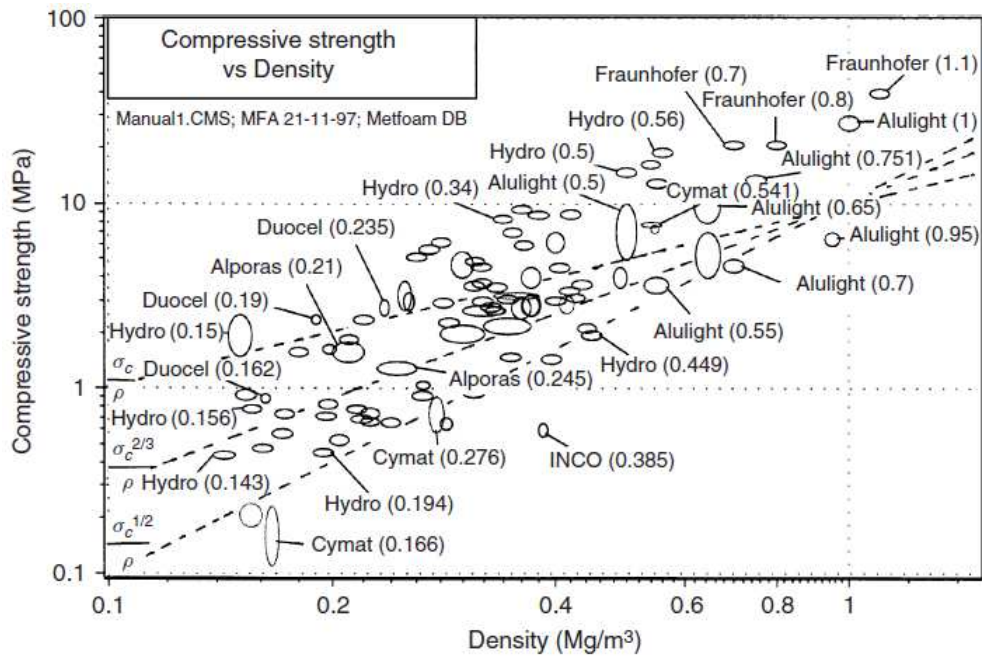
Obrázek 2.3: Závislost hustoty na Youngově modulu [6]

Závislost pevnosti na hustotě

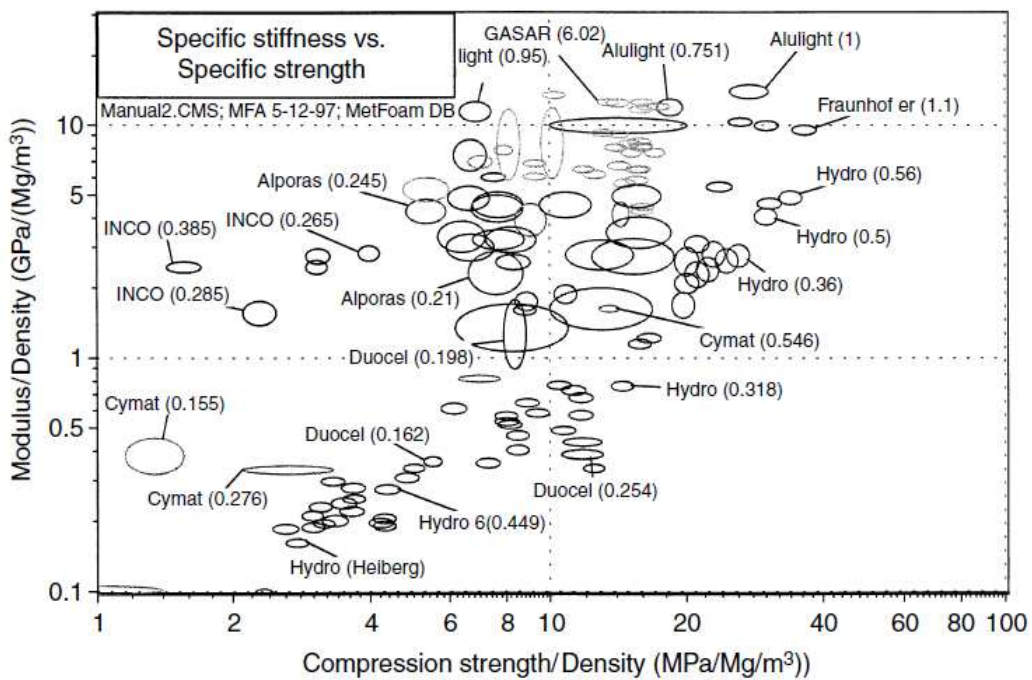
Díky své kombinaci vysoké pevnosti a nízké hustoty, viz obrázek 2.4, se kovové pěny jeví jako vhodné pro použití jako lehké a pevné panely [6].

Závislost tuhost na pevnosti

Vliv tuhosti a pevnosti může být důležitý ve strojním inženýrství, anebo ve stavebnictví. Některé kovové pěny se mohou vyznačovat vhodnějšími kombinacemi těchto vlastností než konvenční materiály [6], viz obrázek 2.5.



Obrázek 2.4: Závislost hustoty na pevnosti v tlaku [6]



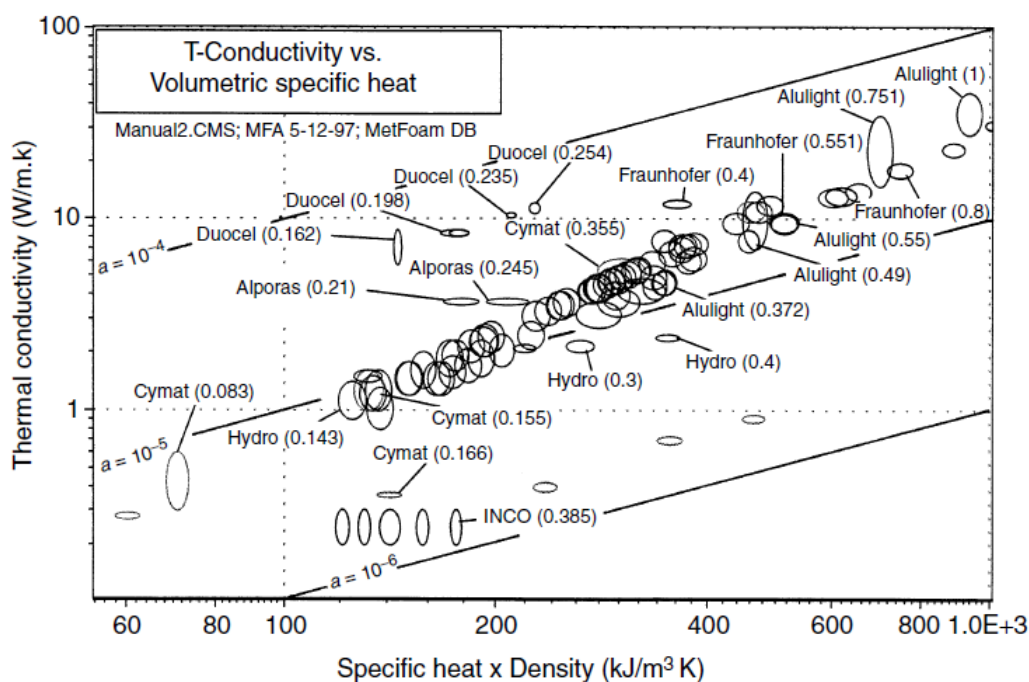
Obrázek 2.5: Závislost pevnosti na tuhosti [6]

2.3.2 Tepelné vlastnosti

Pro kovové pěny platí, že většina jejich tepelných vlastností jako jsou teplota tání, tepelná kapacita a teplotní součinitel roztažnosti jsou stejné jako u materiálů, ze kterých byla pěna vyrobena. Jedinou výjimkou je tepelná vodivost pěny λ , viz obrázek 2.6, pro kterou platí:

$$\lambda \approx \lambda_s \left(\frac{\rho}{\rho_s} \right)^q, \quad (2.3)$$

kde ρ je hustota pěny, ρ_s je hustota materiálu, λ_s je tepelná vodivost materiálu a q nabývá hodnot od 1,65 do 1,8.



Obrázek 2.6: Tepelná vodivost porézních materiálů [6]

2.3.3 Elektrické vlastnosti

Jedinou z elektrických vlastností, kterou má smysl se v případě kovových pěn zabývat je rezistivita R . Pro její velikost platí:

$$R \approx R_s \left(\frac{\rho}{\rho_s} \right)^{-r}, \quad (2.4)$$

kde R je rezistivita pěny, ρ je hustota pěny, r nabývá hodnot od 1,6 do 1,85, R_s je rezistivita materiálu a ρ_s je hustota materiálu.

2.4 Výroba

Technologii výroby kovových pěn lze rozdělit do devíti kategorií. Těmito kategoriemi jsou:

- probublávání plynu skrz roztavené slitiny hliníku s karbidem křemíku, nebo s oxidem hliníkovým,
- vmíchávání „pěnotvorného“ činidla do roztavené slitiny za stálého tlaku a chlazení,
- smíšení kovového prášku s „pěnotvorným“ činidlem, následované ohříváním,
- výroba formy, do které je následně dodán kov v kapalné, nebo práškové formě,
- napařování na polymerní pěnu, která je následně vypálena,
- zachycení vysokotlakého inertního plynu v pórech následované stlačením a následnou expanzí plynu při zvýšené teplotě,
- spečení dutých kuliček,
- slisování kovového prášku s vyluhovatelným práškem a následné louhování, aby zůstala pouze pěnová struktura,
- rozpuštění plynu v zkapalněném kovu [6].

Těchto devět metod lze rozdělit do čtyř následujících tříd:

- pěna je zformována z plynné fáze,
- pěna je získána pomocí elektrolýzy z vodného roztoku,
- pěna je získána zpracováním kapalné fáze,
- pěna je vytvořena v pevném stavu [6].

Každá z těchto metod může být použita k vytvoření porézních materiálů s omezeným rozsahem relativních hustot a velikostí buněk. Některé pěny mají otevřené buňky, ostatní uzavřené [6].

Výsledné vlastnosti kovových pěn jsou ovlivněny jejich složením, relativní hustotou a topologií buněk (otevřené, uzavřené, velikost) [6].

2.5 Využití porézních materiálů

Kovové pěny a obecně i porézní materiály jsou v současné době využívány stále ve větší míře. Finální typ použitého porézního materiálu závisí na požadované poréznosti, tedy na velikosti jednotlivých buněk a na jejich typu. Důležitou roli hrají i výrobní možnosti a ekonomické náklady [4]. Kovové pěny mají využití v mnoha průmyslových odvětvích. Typickým využitím pro porézní materiály jsou například izolační materiály, filtrace a absorpce energie [2]. Ty nejdůležitější budou popsány v následujících odstavcích.

2.5.1 Automobilový průmysl

Při konstrukci moderních automobilů je kladen velký důraz na jejich celkovou bezpečnost, což způsobuje výrazný nárůst jejich hmotnosti. Zvýšená hmotnost naneštěstí vede ke zvýšené spotřebě pohonných hmot, čímž je vyvolána potřeba snížit hmotnost takového automobilu [4].

Částečným řešením těchto problémů je použití kovových pěn. Kovové pěny se vyznačují nízkou hmotností, proto je lze použít pro celkové odlehčení konstrukce. Další velkou výhodou kovových pěn, kterou lze uplatnit v oblasti automobilového průmyslu je jejich schopnost absorpce energie nárazu [4], [12].

2.5.2 Letecký průmysl

Možnost použití kovových pěn v lehkých konstrukcích je v leteckém průmyslu stejně důležitá jako v tom automobilovém. Nahrazením používaných voštinových (honeycomb) struktur jejich ekvivalentem z kovových pěn umožňuje snížení celkových nákladů a v některých případech i zlepšení celkových vlastností. Možností je i použití kovových pěn za účelem zvýšení bezpečnosti [4], [13] [14].

V leteckém průmyslu se kovové pěny rovněž používají i v turbínách, kde je využíváno jejich vysoké tuhosti a schopnosti tlumení [4].

2.5.3 Kosmický průmysl

V kosmonautice jsou kovové pěny využívány ze stejných důvodů jako v letectví. Dále jsou pro svou schopnost tlumit nárazy využívány v přistávacích modulech. Důležité je i jejich využití v nosných konstrukcích, kde nahrazují jiné materiály, které se jen obtížně vyrovnávají s extrémními podmínkami jako jsou třeba velké změny teplot, anebo vakuum, kterým jsou kosmické prostředky vystaveny. Velký význam pro aplikaci v kosmickém průmyslu má i nízká hmotnost tohoto typu materiálu [4], [15].

2.5.4 Námořní průmysl

Stejně jako v předchozích příkladech hraje i v případě stavby lodí velkou roli lehkost celkové konstrukce. V případě stavby lodí je důležité, že pěnové sendvičové struktury mohou být efektivně spojeny, čímž dojde ke zpevnění konstrukce lodě. Při stavbě lodi najdou porézní materiály využití i při konstrukci výtahových plošin, strukturálních přepážek a anténních plošin [4].

2.5.5 Železniční průmysl

Aplikace kovových pěn u železničních vozidel přináší podobné výhody jako v automobilovém průmyslu, stěžejní je zvláště schopnost tlumení nárazu, které se využívá u železničních souprav, u kterých hrozí, že by mohlo dojít ke srážce s automobilem. Porézní materiály se v železničním průmyslu využívají rovněž pro svoji schopnost tlumit hluk [4], [16].

2.5.6 Sportovní vybavení

Navzdory tomu, že se v mnoha sportovních odvětvích uplatňují moderní materiály, u porézních materiálů tomu tak není. Jedním z mála sportů, kde lze porézní materiály využít, je americký fotbal, kde se nabízí použití pro výrobu chráničů a helem. Důvodem k použití porézních materiálů v této oblasti je jejich schopnost energetické absorpce [4].

2.5.7 Biomedicínský průmysl

V současné době se na výrobu protéz používají slitiny titanu, nebo kobaltu s chromem. Důvodem pro použití těchto materiálů je jejich „biokompatibilita“. Stejným způsobem by mělo být možné použít kovovou pěnu, jejíž základ by tvořily některé z uvedených materiálů [4]. Oblast optimalizace náhrad a tkáňových nosičů je v současné době jedno z priorit celosvětového výzkumu [18], [19], [20].

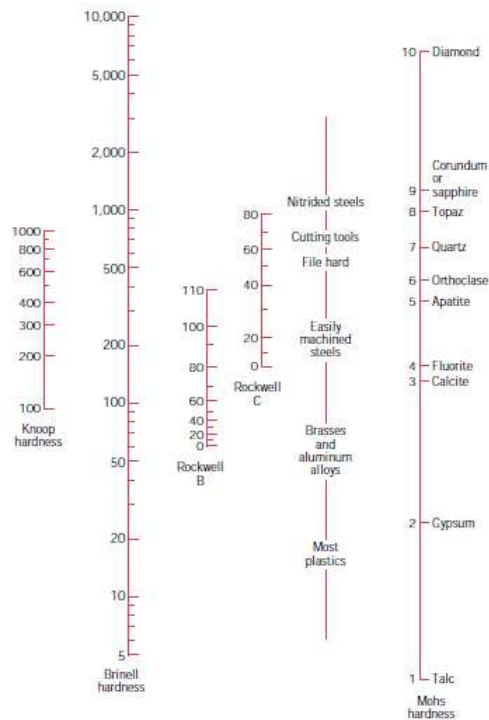
Kapitola 3

Povrchová tvrdost

Tvrdostí materiálu se rozumí jeho vlastnost, která se projevuje jako odpor proti plastické, nebo pružné deformaci, případně jako odpor proti oddělování části povrchu, nebo jako kombinace těchto odporů [21]. Těmito deformacemi může být například malý vpich, nebo vryp [5]. Zkoušky tvrdosti patří mezi nejstarší, ale zároveň i nejpoužívanější zkoušky v oblasti materiálového inženýrství. Zkoušky tvrdosti jsou ve své podstatě zkoušky nedestruktivní, protože narušení, ke kterému na testovaných vzorcích dochází v průběhu testů, je minimální [21].

V současné době existuje více než sto různých metod, které slouží k určení tvrdosti. Těchto sto metod lze rozdělit do dvou základních skupin a to na skupinu metod vnikacích a na skupinu metod odrazových [21]. Nevýhodou existence velkého množství metod určování tvrdosti materiálu je, že porovnávání hodnot tvrdosti je možné pouze v případech, kdy jednak byla k získání porovnávaných hodnot použita stejná metoda a jednak bylo použito pokaždé stejné zkušební zařízení [21]. Převody hodnot mezi různými stupnicemi tvrdosti lze provádět pouze v případě, že existují podklady pro přepočet, příklad srovnání jednotlivých stupnic tvrdosti je na obrázku 3.1. Takovéto podklady se sestavují na základě takzvaných porovnávacích zkoušek [21]. Nejspolehlivější data pro převádění výsledků získaných za pomoci různých metod určování tvrdosti existují pro ocel [5].

Výsledky zkoušek tvrdosti závisí na velkém množství vlivů, z nichž největší význam mají elastické vlastnosti testovaného materiálu, zvláště pak modul pružnosti ve smyku a v tahu, a proto nelze definovat tvrdost jako přesnou fyzikální veličinu. Velkou roli má také velikost síly, kterou je působeno na zkoumaný materiál a rovněž záleží na plastických vlastnostech testovaného vzorku. Plastickými vlastnostmi se v tomto kontextu rozumí zejména mez kluzu a charakter deformačního zpevnování [22]. Výsledná velikost tvrdosti je ovlivněna i tvarem indentoru, rychlostí, kterou



Obrázek 3.1: Srovnání škál tvrdosti [5]

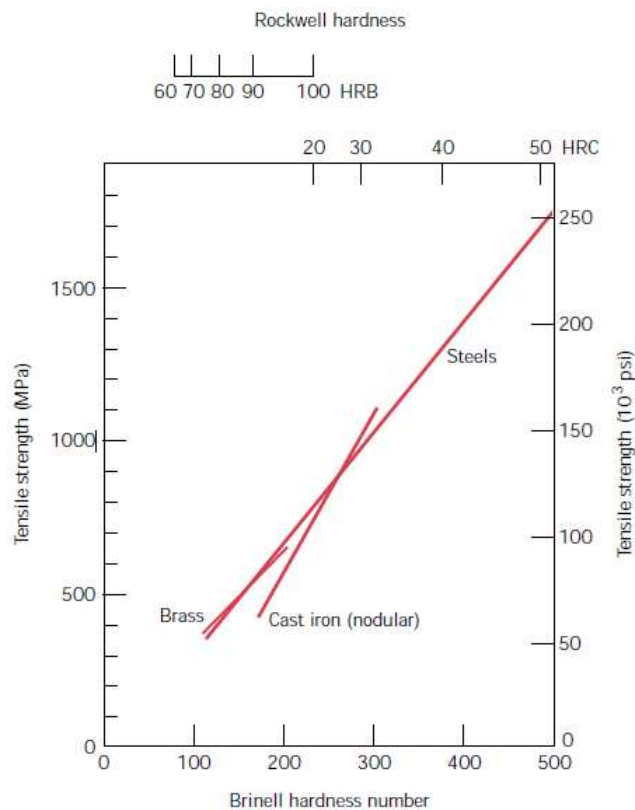
vniká do testovaného tělesa a na konečné hodnotě tvrdosti se projevuje i doba trvání zkoušky. Velkou roli hraje i mez pevnosti [23].

Výsledky zkoušek tvrdosti pro stejný kov se mohou poměrně výrazně odlišovat. Rozdíly mohou být způsobeny na příklad velikostí zrn, přičemž platí, že čím je materiál jemnozrnější, tím vyšší jsou výsledné hodnoty jeho tvrdosti. Velkou roli na výsledné velikosti tvrdosti má i teplota materiálu, neboť s rostoucí teplotou klesá pevnost vazeb mezi atomy. Na tvrdost materiálu má vliv i podíl cizích příměsí, které obvykle snižují schopnost materiálu být plasticky deformován, ale na druhou stranu zvyšují jeho tvrdost. Posledním důvodem různé tvrdosti stejného kovu, který zde bude zmíněn, je vnitřní pnutí, jenž může být způsobeno například nerovnoměrným ochlazováním, anebo tvářením za studena, čímž rovněž dochází ke zvyšování hodnoty tvrdosti [22].

Zkoušky tvrdosti se v porovnání s ostatními zkouškami vlastností materiálů používají výrazně častěji [5]. Důvody proč tomu, tak je jsou shrnuty v následujících několika bodech:

- Zkoušky tvrdosti jsou relativně jednoduché a k jejich provedení není potřeba žádná zvláštní úprava vzorku.

- Testovací vybavení potřebné pro provedení zkoušky tvrdosti je relativně levné.
- Zkoušky tvrdosti jsou relativně nedestruktivní a vzorek je při jejich provádění narušen pouze malou měrou.
- Ostatní vlastnosti testovaného materiálu mohou být odvozeny z výsledků zkoušky tvrdosti, typickým příkladem je třeba pevnost v tahu [5], vztah mezi tvrdostí a pevností v tahu je na obrázku 3.2.



Obrázek 3.2: Vztah mezi tvrdostí a pevností v tahu [5]

3.1 Metody vnikací

Nejrozšířenějšími vnikacími metodami jsou tak zvané statické metody. U těchto metod může být měřítkem tvrdosti buď velikost plastické deformace, anebo velikost elasticko-plastické deformace. Nejpoužívanějšími metodami, které spadají do první skupiny statických metod, jsou metody Brinellovy, Vickersovy a Knoopovy. Typickým zástupcem druhé skupiny statických metod je metoda Rockwellova [21].

U vnikacích metod určování tvrdosti se tvrdostí rozumí odpor testovaného materiálu proti vnikání cizího tělesa. Vnikací tělesa mají většinou tvar nějakého základního geometrického tvaru jako jsou jehlan, koule, nebo kužel [21]. Zpočátku až do dosažené meze úměrnosti je materiál deformován pouze pružně, proto v této fázi o deformaci rozhoduje především modul pružnosti testovaného materiálu. Z tohoto důvodu není v oblasti pružných deformací tvrdost materiálovou vlastností. Na tuto skutečnost je třeba dávat pozor především při zkouškách velmi tvrdých materiálů a u zkoušek, kdy je používána velmi malá zatěžovací síla [23].

Výsledná velikost tvrdosti tedy vyjadřuje dvě vlastnosti testovaného materiálu. První z nich je odpor vůči deformaci pružné a druhý z nich je odpor proti plastické deformaci. V případě vnikacích zkoušek tvrdosti převažuje odpor proti plastickým deformacím [23].

3.1.1 Brinellova metoda

Základní myšlenkou této zkoušky je vtlačování vnikacího tělesa s tvarem koule o průměru D , které je obvykle vyrobeno z karbidu wolframu [24], silou F do povrchu zkoumaného materiálu. V minulosti se pro zkoušky tvrdosti podle Brinella používala i vnikací tělesa z oceli nebo wolframu [21]. Po zatlačení vnikacího tělesa následuje odlehčení vnikacího tělesa a změření průměru vtisku d , viz obrázek 3.3. Průměr vtisku se získá vypočtením průměru ze dvou na sebe kolmých velikostí průměru vtisku. Pro určení velikosti průměru se používají optické metody manuální, anebo automatické [24]. Velikost tvrdosti podle Brinella HB je funkcí zatížení a průměru vtisku [5]. Výsledná tvrdost se určuje podle následujícího vztahu:

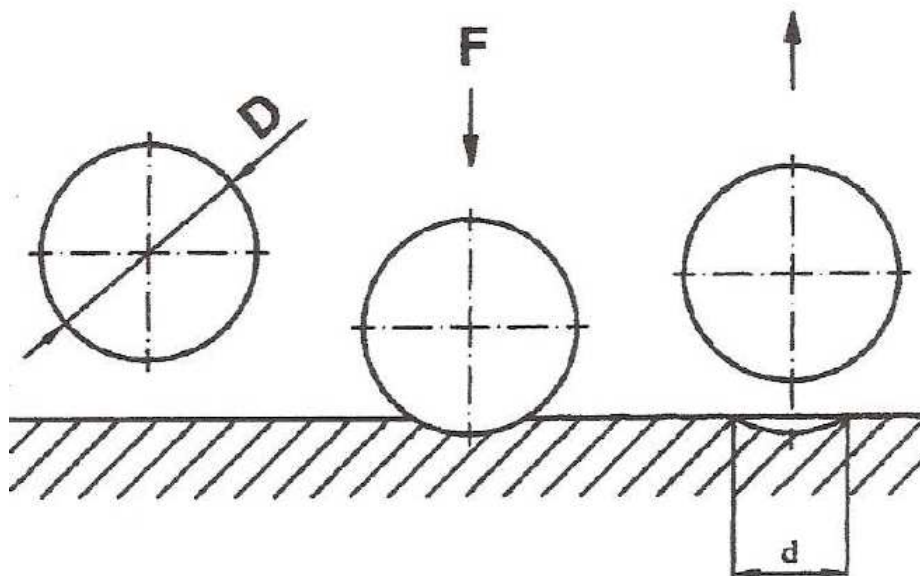
$$HB = 0,102 \frac{2F}{\pi D(D - \sqrt{D^2 - d^2})}, \quad (3.1)$$

kde F je zkušební síla v Newtonech, D je průměr kuličky v milimetrech a d je průměr v tisku v milimetrech [21]. Výsledek měření tvrdosti podle Brinella je zapisován v následujícím tvaru:

$$600HB1/30/20, \quad (3.2)$$

kde první číslo označuje velikost tvrdosti vypočtenou podle vzorce, který byl představen výše, zkratka HBW popisuje typ zkoušky. Lze se setkat i se značením HBW , nebo HBS , kde třetí písmeno ve značce uvádí materiál vnikacího tělesa - W označuje wolfram a S ocel. Číslo 1 udává průměr vnikacího tělesa v milimetrech. Číslo 30 značí použité ekvivalentní zkušební zatížení v kilopondech a poslední hodnota označuje dobu trvání zkoušky v sekundách. Poslední hodnota

se uvádí pouze v případech, kdy doba trvání zkoušky neležela v doporučeném intervalu 10 – 15 s [24].



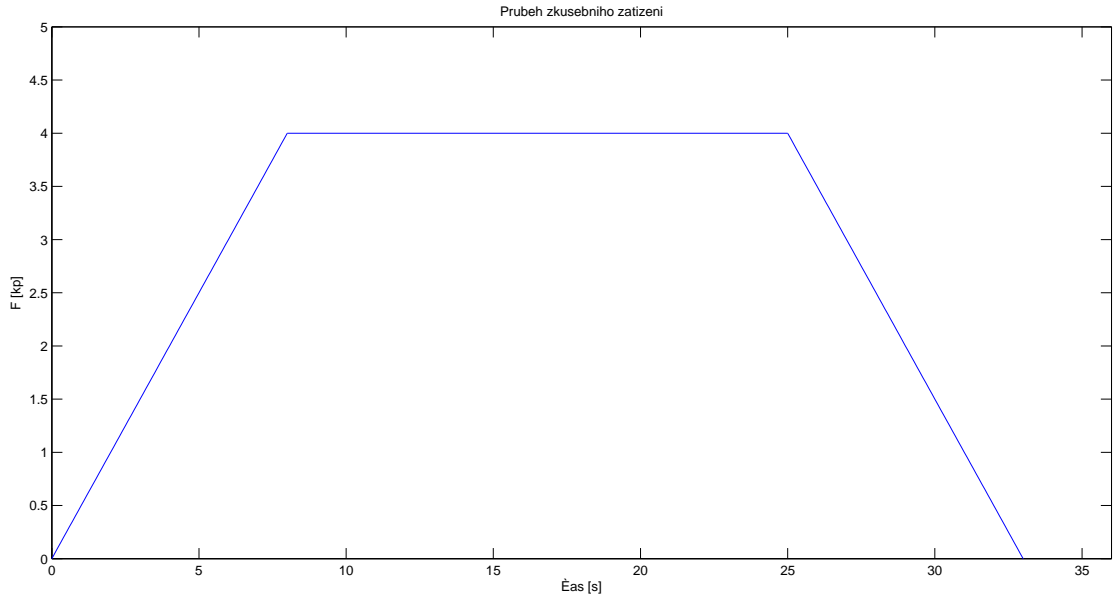
Obrázek 3.3: Průběh zkoušky tvrdosti podle Brinella [23]

Při provádění zkoušek tvrdosti podle Brinella je nutné, aby byl testovaný povrch hladký, rovný, bez okujené vrstvy, bez cizích tělísek a rovněž úplně bez mazadel. Rovněž je nutné, aby byl povrch testovaného materiálu ovlivněn v co nejmenší míře ohřevem a tvářením. Testovaný vzorek musí být minimálně osmkrát tak tlustý než činná hloubka vtisku vnikacího tělesa [24]. Po provedení zkoušky se nesmí na protilehlém povrchu testovaného tělesa projevit žádné stopy deformace [21].

Zkouška tvrdosti podle Brinella se provádí při teplotách od 10 do 35 °C, přičemž ideálně je pohybovat se kolem teploty 23 ± 5 °C [24]. Použité zkušební zařízení musí být vybráno tak, aby odpovídalo zkoušenému materiálu a velikosti kuličky. Během zkoušky musí být vzorek položen na tuhé podložce. Je nutné, aby byl vzorek uložen tak, aby nemohlo dojít k jeho pohybu [21].

Samotná zkouška probíhá tak, že vnikací těleso je zatlačováno do povrchu vzorku takovým způsobem, že zatížení působí kolmo k jeho povrchu. Vnikací těleso není během zkoušky vystaveno žádným rázům nebo chvění. Doba mezi začátkem testování a dosažením požadované zkušební síly musí být větší než dvě sekundy a kratší než osm sekund, viz obrázek 3.4. Po dosažení zkušební síly je testované těleso vystaveno jejímu působení po dobu deseti až patnácti sekund. Ve výjimečných případech mohou být testované vzorky vystaveny zkušební síle i po delší dobu [21].

Při provádění zkoušky je třeba dát pozor na to, aby byl střed vtisku v patřičné vzdálenosti



Obrázek 3.4: Průběh zkušebního zatížení během provádění zkoušky podle Brinellovy metody [21]

od okraje vzorku. Velikost této vzdálenosti závisí na typu testovaného materiálu. V případě, že se jedná o ocel, litinu, měď, případně o její slitiny musí mít tato vzdálenost velikost dva a půl násobku průměru vtisku. Pokud je testovaným materiálem nějaký z lehkých kovů, olovo, cín, nebo jejich slitina, pak musí být vzdálenost od středu vtisku k okraji trojnásobek průměru vtisku [21].

Rovněž existují pravidla pro vzdálenosti mezi dvěma sousedními vtisky. Pro první skupinu materiálů definovanou v předchozím odstavci musí být mezi jednotlivými vtisky vzdálenost odpovídající čtyřnásobku průměru vtisku a pro druhou skupinu se jedná o šestnásobek průměru vtisku [21].

Zkouška tvrdosti podle Brinella je nejvhodnější pro určování tvrdosti měkkých a středně tvrdých materiálů. V případě tvrdých materiálů dochází k výrazné deformaci testovacího tělesa, a proto je vhodnější použít jinou metodu. Velkou výhodou zkoušky podle Brinella je, že je otestována průměrná tvrdost z relativně velké plochy. Nevýhodou pak je, že tento vtisk je relativně velký a narušuje tak vzhled větší části povrchu [23].

3.1.2 Vickersova metoda

V případě Vickersovy metody je vnikacím tělesem diamantový čtyřboký jehlan, který má čtvercovou základnu. U tohoto vnikacího tělesa je daný vrcholový úhel o velikosti 136° , který mezi sebou svírají protilehlé stěny. Po vykonání zkoušky je měřena velikost úhlopříčky vtisku, který vznikne po odlehčení, viz obrázek 3.5. Tvrdost podle Vickerse je definována jako poměr zkušebnímu zatížení ku velikosti povrchu vtisku [21]. Zkouška tvrdosti podle Vickerse se označuje dvojitými písmeny HV za níž následuje číslo udávající velikost zkušebnímu zatížení. V případě, že zkouška probíhala po nestandardní dobu, je součástí značení i doba trvání zkoušky [22]. Velikost výsledné tvrdosti je pak dána následujícím vztahem:

$$HV = 0,102 \frac{2F \sin \frac{136^\circ}{2}}{d^2} = 0,1891 \frac{F}{d^2}, \quad (3.3)$$

kde F představuje zkušební zatížení v Newtonech a d je aritmetický průměr velikostí úhlopříček vtisku v milimetrech [21]. Výsledek měření tvrdosti podle Vickerse se zapisuje v následujícím tvaru:

$$640HV30/20, \quad (3.4)$$

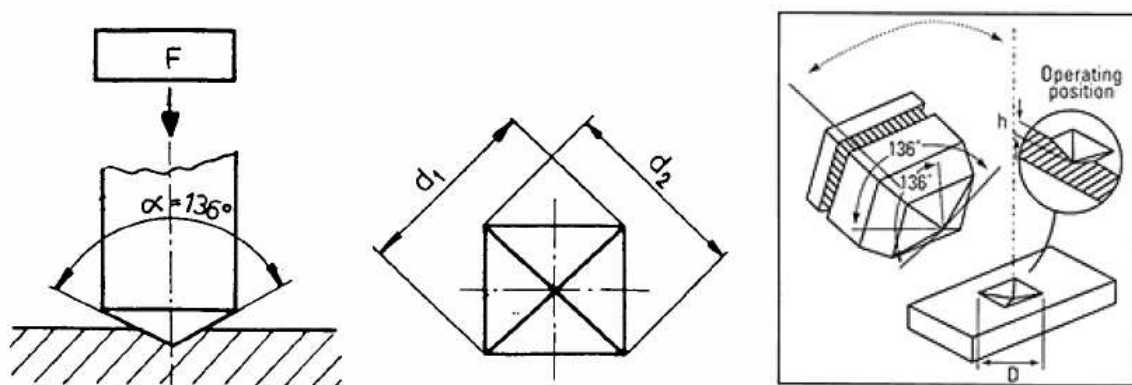
kde první hodnota udává velikost tvrdosti podle Vickerse, která se vypočítá podle vztahu, který byl zmíněn výše. Dvojice písmen HV označuje způsob měření tvrdosti. Předposlední číslice udává velikost ekvivalentního zkušebnímu zatížení v kilopondech. Poslední hodnota udává dobu trvání zkoušky tvrdosti v sekundách, tato hodnota se uvádí pouze v případě, kdy doba trvání zkoušky nespadá do intervalu 10 – 15 s [25].

Vnikací zkoušky postavené na Vickersově metodě jsou na základě použitého zkušebnímu zatížení členěny do tří kategorií:

- zkouška tvrdosti podle Vickerse,
- zkouška tvrdosti podle Vickerse při nízkém zatížení,
- zkouška mikrotvrdosti podle Vickerse [21].

Pro provádění zkoušky tvrdosti podle Vickerse platí obdobná pravidla jako pro zkoušky podle Brinella. Testovaný povrch musí být hladký, rovný, bez okujené vrstvy, bez cizích tělísek a bez mazadel. Tento povrch musí být rovněž upraven takovým způsobem, aby bylo možné

přesně stanovit velikosti úhlopříček vtisku. Vzorky musí být připraveny tak, aby byly co nejméně změněny jejich vlastnosti, k čemuž by mohlo dojít například vlivem ohřevu [21]. Tloušťka testovaného vzorku musí činit nejméně jeden a půl násobek délky úhlopříčky vtisku [25]. Rovněž se na protilehlém povrchu testovaného materiálu nesmí projevit žádné stopy deformace [21]. Zkouška tvrdosti podle Vickerse se stejně jako v případě zkoušky tvrdosti podle Brinella provádí při teplotách od 10 do 35 °C, přičemž ideální je pohybovat se kolem teploty 23 ± 5 °C [25].



Obrázek 3.5: Průběh zkoušky tvrdosti podle Vickerse [22]

Pravidla pro vzdálenost mezi jednotlivými vtisky a vzdálenost mezi okrajem a vtiskem se odvíjí od testovaného vzorku. Pokud je testována tvrdost oceli, mědi, případně slitin mědi, tak vzdálenost mezi středy jednotlivých vtisků musí činit tři násobek velikosti úhlopříčky a vzdálenost mezi středem vtisku a okrajem vzorku musí být dvou a půl násobek velikosti úhlopříčky vtisku [25].

Pokud zkoumáme velikost tvrdosti lehkých kovů, olova, cínu, případně jejich slitin, pak vzdálenost mezi středy jednotlivých vtisků musí být šesti násobek velikosti úhlopříčky a vzdálenost mezi středem vtisku a okrajem testovaného vzorku musí být tří násobek velikosti úhlopříčky [25].

Pokud se na jednom vzorku materiálu nachází vtisky s různými velikostmi, pak se jako určující velikost při určování jejich minimální vzdálenosti bere ten s větší úhlopříčkou [25].

Největší výhodou zkoušky tvrdosti podle Vickerse je, že je ze všech používaných vnikacích metod nejpřesnější a je použita pro drtivou většinu všech možných materiálů. Další výhodou je relativně malý vtisk, který zůstává po provedení zkoušky. S relativně malou velikostí vtisku souvisí i sporná použitelnost pro hrubozrné a nehomogenní materiály, kdy nelze získat jednoznačné výsledky. Na druhou stranu lze díky malé velikosti vtisku přesně měřit tvrdost jednotlivých částí nehomogenních materiálů. Nevýhodou představuje i nutnost náročnější přípravy zkoušky

a nutnost přesného měření úhlopříček [23].

3.1.3 Rockwellova metoda

Rockwellova zkouška představuje nejpoužívanější metodu určování tvrdosti materiálů [5]. Při této zkoušce se mohou používat indenty různých průměrů (1,588, 3,175, 6,350 a 12,17 mm) a tvarů. Použité indenty mají tvar buď kuličky a nebo jehlanu s vrcholovým úhlem 120° . Kulička může být vyrobena z wolframu, anebo z oceli, jehlan bývá diamantový [26]. Rovněž je možné používat zatížení o rozdílných velikostech [5]. Zkouška probíhá na specializovaném zařízení, viz obrázek 3.6 tak, že nejdříve je indenter vtlačován do testovaného materiálu pod předběžným zatížením. Hloubka, které je při tomto zatížení dosaženo je zvolena jako výchozí. Následně je k předběžnému zatížení přidáno přídatné zatížení. Po určité době je přídatné zatížení odstraněno, poté je změřen rozdíl výchozí a konečné hloubky, viz obrázek 3.7. Je nutné se vyvarovat rázů a chvění dokud není dosaženo hodnoty předběžného zatížení. Doba náběhu a působení přídatného zatížení se liší podle typu testovaného materiálu [21]. Průběh zkušebního zatížení je znázorněn na obrázku 3.8. Výsledná tvrdost se spočítá podle následujícího vzorce:

$$HR = N - \frac{h}{S}, \quad (3.5)$$

kde h je rozdíl výšek změřený po prvním a po druhém zatížení. N a S jsou konstanty. Velikost konstanty S je 0,002 mm. Velikost konstanty N se odvíjí od použité stupnice tvrdosti [26]. Velikost výsledné tvrdosti podle Rockwella se zapisuje v následujícím tvaru:

$$70HR30TW, \quad (3.6)$$

kde první číslice udává výslednou tvrdost, HR značí, že se jedná o zkoušku tvrdosti podle Rockwella, $30T$ označuje typ Rockwellovy škály a poslední písmeno udává materiál, z kterého je vnikací těleso vyrobeno [26].

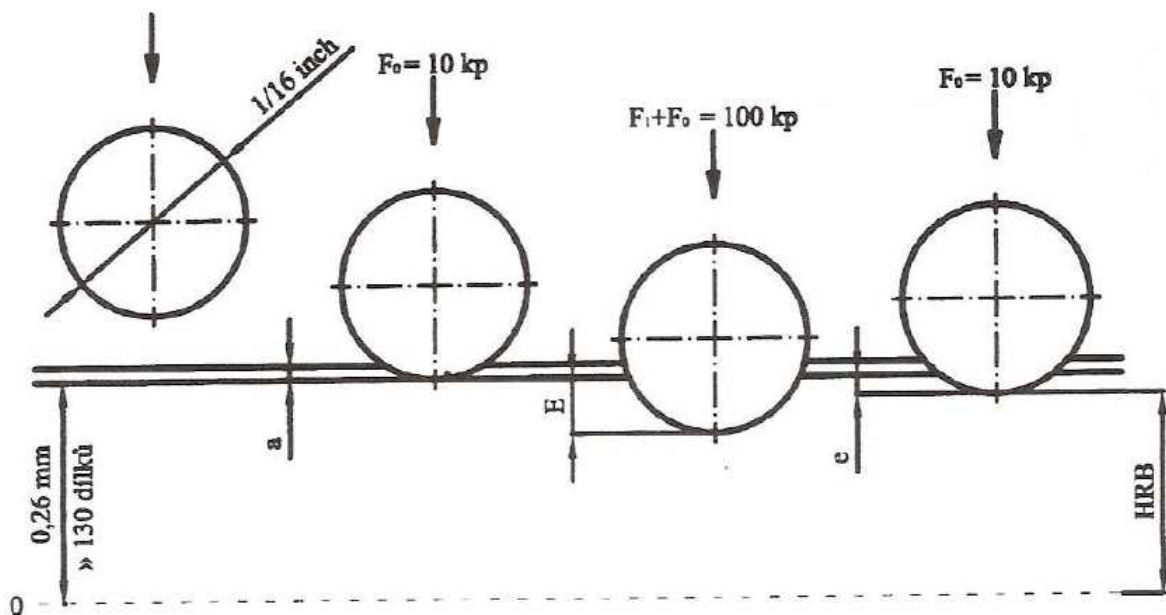
Povrch, na kterém se provádí zkouška tvrdosti podle Rockwella, musí být hladký, rovný, bez okujené vrstvy, bez cizích tělísek a bez mazadel. Jedinou výjimkou z tohoto pravidla je titan, u kterého může být použito mazadlo [21].

Při přípravě testovaného vzorku je třeba brát ohled na to, aby nedošlo k žádné změně jeho vlastností. K těmto změnám by mohlo dojít například kvůli ohřevu, řezání, broušení, anebo leštění. Testovaný vzorek musí mít takovou tloušťku, aby její velikost činili minimálně desetinásobek



Obrázek 3.6: Zařízení pro zkoušku tvrdosti podle Rockwella [5]

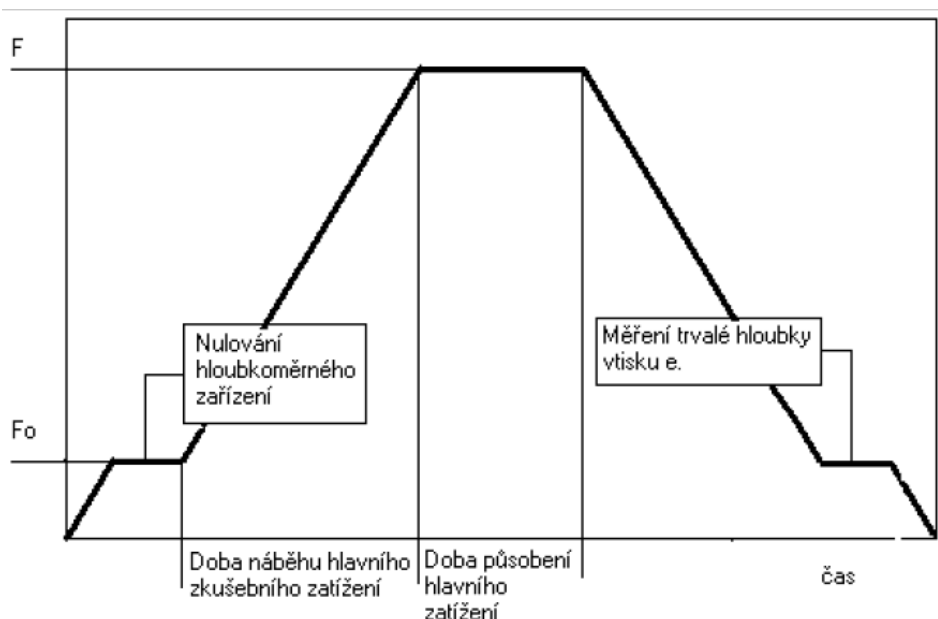
e. Zároveň je nutné, aby po provedení zkoušky tvrdosti nebyly žádné patrné stopy deformace na protilehlém povrchu testovaného materiálu [21]. Zkouška se provádí za teplot, které spadají do intervalu 10 do 35 °C [26].



Obrázek 3.7: Průběh zkoušky tvrdosti podle Rockwella [23]

Ohledně teploty, při které je zkouška prováděna a uložení vzorku platí stejná pravidla jako v případě zkoušek tvrdosti podle Brinella a Vickerse [21].

Vzdálenost mezi středy jednotlivých vtisků musí být minimálně tři násobek velikosti úhlopříčky / průměru a vzdálenost mezi středy vtisku a hranou testovaného materiálu musí být dvou a půl násobek velikost úhlopříčky / průměru vtisku [26].



Obrázek 3.8: Průběh zkušebního zatížení během měření tvrdosti podle Rockwellovy metody [21]

Největší výhodou zkoušky tvrdosti podle Rockwella je její značná jednoduchost. Rockwellova metoda je nejvhodnější pro automatizované zkoušky využívané v průmyslové praxi. Lze ji používat jak pro měkké, tak i pro tvrdé materiály. Její nejvýraznější nevýhodou je, že nemůže být použita pro určování tvrdosti velmi tenkých plechů a pásů [23].

3.1.4 Knoopova metoda

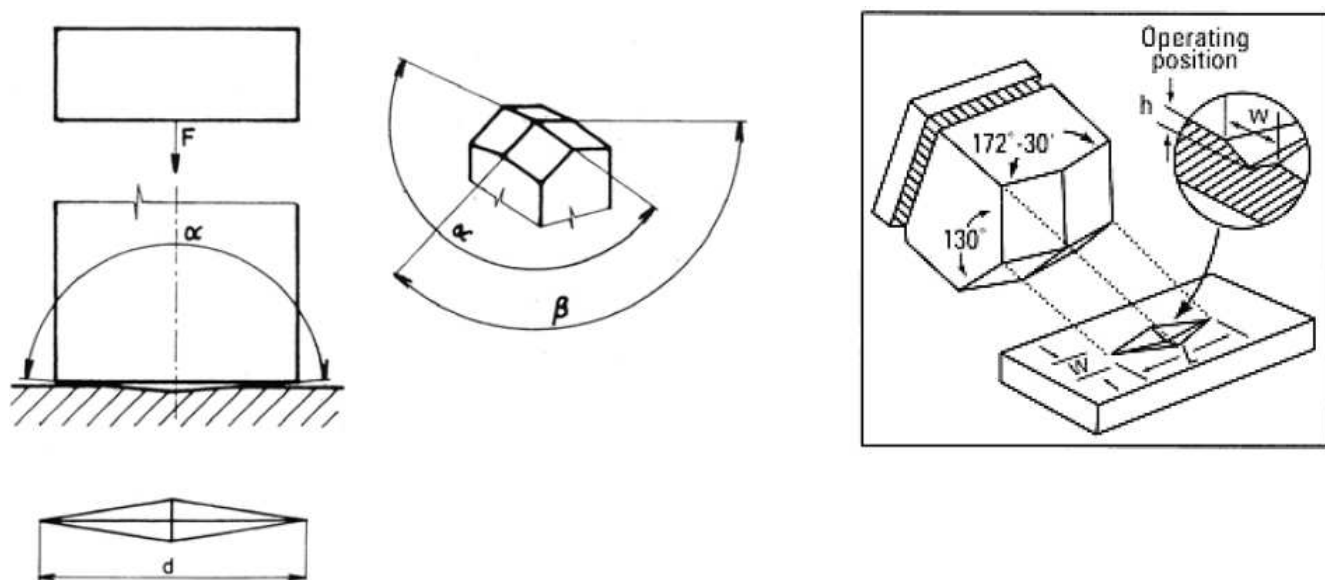
Jedna z méně používaných metod, které se užívá k určování tvrdosti materiálů je metoda Knoopova. Tato metoda je určena pro měření tvrdosti u velmi tenkých vrstev. Podobně jako u Vickersovy metody je i zde vnikacím tělesem diamantový čtyřboký jehlan. Rozdíl oproti Vickersově metodě spočívá v jiném tvaru hrotu, viz obrázek 3.9. Pro další výpočty tvrdosti se měří pouze velikost větší úhlopříčky [21]. Velikost tvrdosti podle Knoopa se označuje dvojicí písmen HK [22]. Výsledná velikost tvrdosti podle Knoopa se vypočítá podle následujícího vztahu:

$$HK = \frac{1,4509F}{d^2}, \quad (3.7)$$

kde F je velikost působícího zatížení v Newtonech a d představuje velikost větší s dvojce úhlopříček v milimetrech [21]. Výsledek měření velikost tvrdosti podle Knoopa se zapisuje v následujícím tvaru:

$$640HK0,1/20, \quad (3.8)$$

kde první číslo udává výslednou velikost tvrdosti, HK určuje typ zkoušky tvrdosti, pomocí níž bylo měření prováděno. 0,1 představuje přibližnou velikost ekvivalentního zatížení v kilopondech a poslední hodnota udává čas v sekundách, po který zkouška tvrdosti probíhala. Tato hodnota se uvádí pouze v případech, kdy čas měření nespadá do obvyklého intervalu 10 – 15 s [27].



Obrázek 3.9: Knoopova metoda [22]

Pro vlastnosti povrchu testovaného materiálu a jeho přípravu platí stejné podmínky jako u předchozích zmíněných metod určování tvrdosti [27]. Zkouška by měla probíhat za teplot kolem $23 \pm 5^\circ\text{C}$ [27].

Pro vzdálenost středů dvou sousedních vpichů platí, že mezi nimi musí být vzdálenost minimálně dvou a půl násobku velikosti kratší úhlopříčky, pokud jsou orientované vedle sebe a jedna dlouhá úhlopříčka, pokud jsou orientovány za sebou. Vzdálenost mezi středem vpichu a hranou testovaného vzorku musí být minimálně tří násobek kratší úhlopříčky [27].

3.1.5 Metoda universální tvrdosti

Universální metoda měření tvrdosti spočívá ve vnikání diamantového tělesa, které má stejný tvar jako Vickersův hrot. Během testování je průběžně měřena velikost zkušební síly F a hloubka vtisku h . Universální tvrdost se vypočítá dosazením do následujících vztahů:

$$A(h) = \frac{4 \sin \frac{\alpha}{2}}{\cos^2 \frac{\alpha}{2}} h^2, \quad (3.9)$$

$$HU = \frac{F}{A(h)} = \frac{F}{26,43h^2}, \quad (3.10)$$

kde F je působící síla v Newtonech, h je hloubka vtisku v milimetrech a α je vrcholový úhel o velikosti 136° [21].

U této metody se předpokládalo, že nahradí všechny ostatní metody určování tvrdosti, proto její označení jako „universální“, tento předpoklad se však nenaplnil [21].

Kapitola 4

Teorie zpracování obrazových dat

Zpracování obrazových dat, které je v zahraniční literatuře označováno jako Image processing se používá ke dvěma cílům. Prvním z nich je zlepšení vzhledu obrazu, za účelem zlepšení jeho čitelnosti pro uživatele, druhým pak je taková úprava obrazu, která umožní měření vlastností [28].

Metody zpracování obrazových dat lze rozdělit do několika základních kategorií. Třídícími kritérii jsou v tomto případě vstupy a výstupy do procesu a operace, které v rámci zpracování obrazu probíhají. Těmi základními kategoriemi jsou:

- Low-level - v tomto případě jsou jak vstupními, tak i výstupními daty obrazy,
- Mid-level - na této úrovni jsou vstupy obrazy a výstupy jsou parametry, které byly získány během zpracování těchto vstupů. Výstupy tak mohou být třeba hrany, nebo identifikované objekty. Procesy, které zde probíhají, jsou na příklad segmentace, nebo klasifikace objektů,
- High-level - v případě High-levelu jsou stejně jako v předchozích případech vstupem obrazová data, výstupem pak jsou, po provedení obrazové analýzy, parametry s přiřazeným významem [29].

Existuje velké množství metod a operací, pomocí kterých lze zpracovávat obrazová data. V této kapitole bude věnována pozornost pouze těm, která mají svůj význam při plnění cílů této diplomové práce.

4.1 Pořízení obrazu

Existuje velké množství různých způsobů pro snímání obrazu. Mezi sebou se odlišují jednak konstrukcí a jednak fyzikálním jevem na základě, kterého snímají obraz [30]. V závislosti na

použitém typu snímacího zařízení se liší výsledná kvalita obrazu. Každý typ snímacího zařízení je navíc vhodný pro snímání jiných obrazových dat [28]. Všechna obrazová data, která jsou zpracovávána v rámci této diplomové práce byla pořízena přístrojem s označením MANTA G-504B od německé společnosti AVT. Tento přístroj disponuje rozlišením 2452×2056 pixelů.

4.2 Chyby obrazu

Při snímání obrazových dat může dojít k tomu že nasnímané obrázky vykazují některé nevhodné vlastnosti, které brání ve vhodném zpracování obrazu. Mezi takovéto vady obrazu patří například šum na obraze, nebo nevhodné nasvícení.

4.3 Segmentace a Thresholding

Jednou z nejpoužívanějších metod, jak získat z obrazových dat užitečné informace, je segmentace. Segmentací se rozumí rozčlenění obrazu do částí [28].

4.3.1 Thresholding

Thresholding, který je v českojazyčné literatuře také označován jako prahování, je proces, který slouží ke konverzi původního obrazu, jehož jednotlivé pixely mají různé odstíny šedi, do obrazu černobílého. Základem prahování je šedého pixelu, který si určíme jako referenční. Na základě tohoto zvoleného pixelu část pixelů získá hodnotu logická 0 a část logická 1. Existují čtyři způsoby jak určit, které pixely budou černé a které bílé [31]:

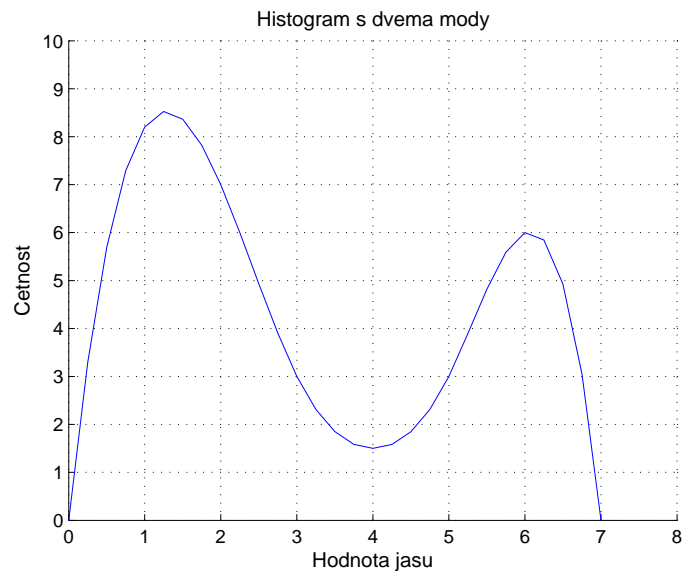
- Threshold above,
- Threshold below,
- Threshold inside,
- Threshold outside [30].

Threshold above znamená, že všechny pixely, které mají větší hodnotu než referenční pixel se stávají pixely popředí a ostatní se stávají pixely pozadí. V případě Thresholdu below je tomu přesně naopak. Pro Threshold inside jsou potřeba dva referenční pixely, tzv. Threshold lower a upper. Následně jsou všechny pixely, které svou hodnotou spadájí mezi threshold lower a upper označeny jako pixely popředí a ostatní jako pixely pozadí. V případě Thresholdu outside je tomu naopak [30].

Samotný fakt, že víme, které šedé pixely se stanou bílými a které černými, nám bohužel nic neříká o tom jak získáme potřebnou hodnotu Threshold levelu. Pro určení velikosti hodnoty Thresholdu existuje několik základních metod. Těmi nejpoužívanějšími jsou metoda histogramů a Otsuova metoda [30]. Další možností jak zvolit velikost prahu je na základě pozorování uživatele [28].

Metoda histogramů

Prvním krokem pro vykonání této metody je vytvoření histogramu z obrazu v odstínech šedi. Vytvořený histogram má dva peaky, viz obrázek 4.1, s tím že jeden s těchto peaků je pro světlé pixely a druhý pro tmavé. Hledaným threshold levelem, tedy prahem, se tedy může stát jakýkoliv pixel ležící mezi těmito dvěma módy. Velkou výhodou metody histogramů je možnost ji relativně snadno zautomatizovat [30].



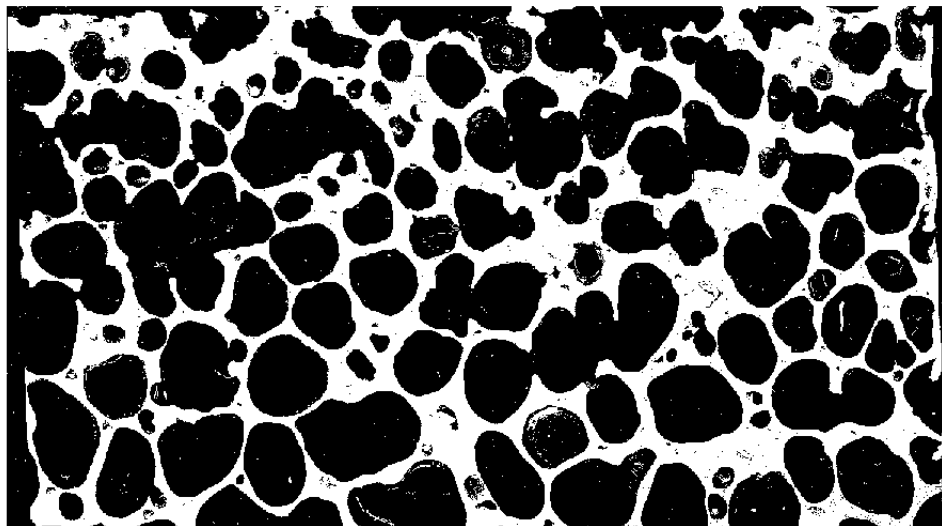
Obrázek 4.1: Histogram s dvěma módy [30]

Otsuova metoda

Otsuova metoda stojí na minimalizaci odchylek dvou skupin pixelů, které jsou separovány referenčním pixelem. Hodnota tohoto referenčního pixelu, tedy prahu, se určuje pomocí iterativního procházení všech hodnot, které by teoreticky mohly být prahem, zatímco je porovnáván rozptyl intenzit pixelů popředí a pozadí. Cílem je nalézt takový referenční pixel, u kterého je rozptyl mezi intenzitami pixelů a pozadí nejmenší [30].

4.4 Binární obraz

Binární obraz je takový obraz, který má pouze černé a bílé pixely, respektive pixely s binárními hodnotami nula a jedna. Příklad takového binárního obrazu je v obrázku 4.2. Postup jakým získat z původního barevného obrazu binární obraz byl popsán v předchozích kapitolách. Nevýhodou konverze do binárního obrazu je, že vždycky bude existovat určitý podíl pixelů, který byl převeden nevhodně. Tyto pixely se obvykle nachází na rozhraní rozdílných vzorů [28].



Obrázek 4.2: Binární obraz

4.5 Základní morfologické operace

K upravení binárního obrazu takovým způsobem, abychom získali obrazová data, která jsou vhodná pro další zpracování, se používají morfologické operace. Jsou to takové operace, jejichž cílem je usnadnit určení parametrů jednotlivých objektů jako třeba hranice, nebo tvary. Základem všech morfologických operací jsou dílčí operace, které jsou označovány jako dilatace, eroze, otevření a uzavření [29].

Všechny morfologické operace spočívají ve výpočtu výsledku matematické operace, která probíhá mezi binárním obrazem B a strukturním elementem S . Strukturní element S bývá typicky výrazně menší než binární obraz. Tento element funguje jako maska. Obvykle má tvar čtyřúhelníku, anebo kruhu, ale teoreticky je možné i použití jiných tvarů [30].

4.5.1 Otevření

Operace označovaná jako otevření, slouží k likvidaci malých oblastí, které zůstanou mimo hlavní oblastí, viz obrázek 4.3. Otevření v sobě zahrnuje erozi, po které následuje dilatace s tím, že je pokaždé použit strukturní element S se stejnou velikostí [29]. Otevření lze popsat následující rovnicí [30]:

$$B \circ S = (B \ominus S) \oplus S. \quad (4.1)$$



Obrázek 4.3: Obrazová data před a po provedení operace Otevření

4.5.2 Uzavření

Uzavření slouží k vyplnění prázdných prostorů uvnitř objektů a ke spojení malých objektů do větších struktur, viz obrázek 4.4. Uzavření se skládá z dilatace následované erozí. Stejně jako v případě otevření probíhá celý proces podle jednoho strukturního elementu S [29]. Pro popis Uzavření slouží tento vztah [30]:

$$B \bullet S = (B \oplus S) \ominus S. \quad (4.2)$$

4.5.3 Dilatace

Během dilatace je prováděn logický součet mezi body binárního obrazu a strukturního elementu [29]. Dilatace je popsána následující rovnicí [30]:



Obrázek 4.4: Obrazová data před a po provedení operace Uzavření

$$B \oplus S = \bigcup_{b \in B} S_b. \quad (4.3)$$

4.5.4 Eroze

Během eroze je naopak prováděn logický rozdíl mezi body binárního obrazu a strukturního elementu [29]. Eroze je pak popsána následujícím vztahem [30]:

$$B \ominus S = \{b \mid b + s \in B \forall s \in S\}. \quad (4.4)$$

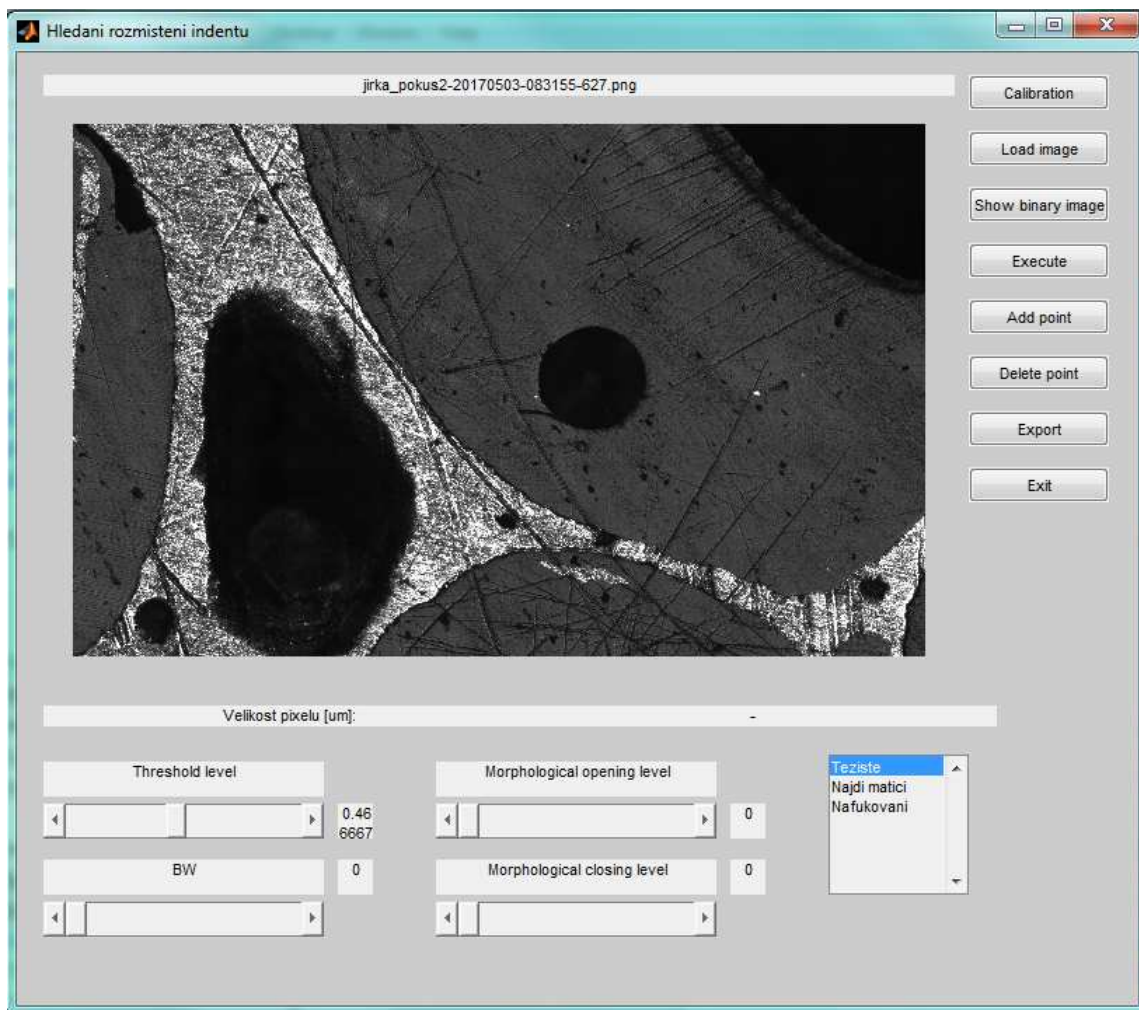
Kapitola 5

Vývoj nástroje pro zpracování obrazových dat

V rámci této kapitoly je řešen jeden z dílčích úkolů této diplomové práce, tedy rozmístění indentů na základě zpracování obrazových dat testovaného vzorku a časová optimalizace této procedury. Pro vytvoření této aplikace byl zvolen softwarový nástroj Matlab od společnost MathWorks, Inc, spolu s balíkem nástrojů Image Processing Toolbox. Tento softwarový balík obsahuje předpřipravené algoritmy pro zpracování obrazu, analýzu obrazových dat, odstraňování chyb a registraci objektů [32]. Důvodem pro zvolení Matlabu na úkor rovněž navrhovaného jazyka Python, byla zkušenost s jeho použitím v rámci mé bakalářské práce [31].

Po spuštění programu se uživateli otevře grafické uživatelské rozhraní, viz obrázek 5.1. Z důvodů větší kontroly nad vytvářením uživatelského rozhraní, nebyl při vytváření použit předpřipravený průvodce. V levé horní části okna se nachází prostor pro grafický výstup činnosti programu. Zde se zobrazuje zpracovávaný obraz a zde jsou v pozdějších fázích k vidění i navrhovaná umístění indentů. V pravé horní části okna se nachází menu, které bude podrobněji popsáno níže. V levé dolní části okna se nachází posuvníky, které ovlivňují způsob jakým bude provedeno zpracování obrazu. V poslední, pravé dolní, části okna se nachází nabídka tří procedur, které lze využít k nalezení nejvhodnějších míst pro indent. Princip fungování těchto tří procedur bude podrobněji popsán v podkapitole 5.3.

V následujících podkapitolách budou popsány principy fungování jednotlivých částí programu.

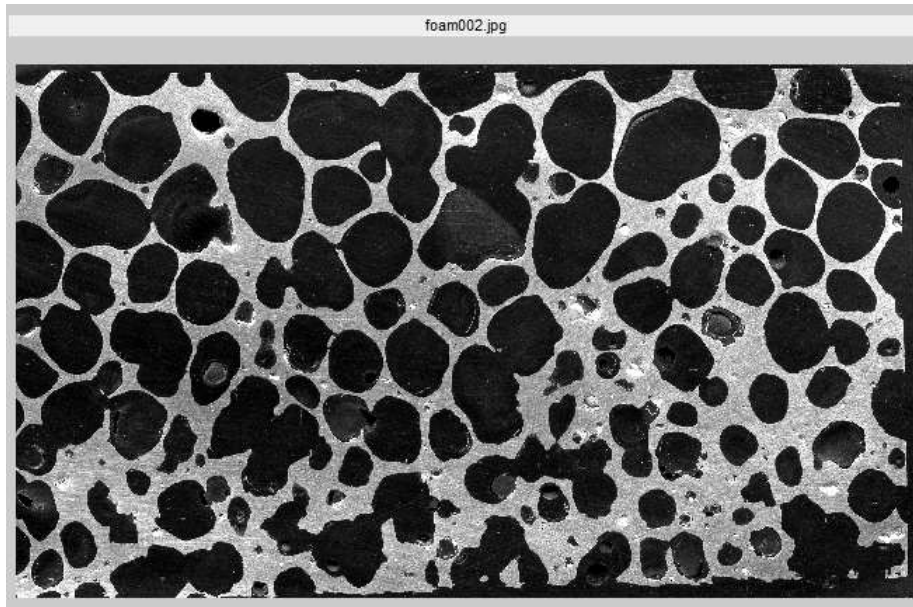


Obrázek 5.1: Grafické uživatelské rozhraní

5.1 Grafický výstup

V této části programu, viz obrázek 5.2, se uživateli zobrazují zpracovávaná data ve všech fázích. Po načtení zdrojových dat se zde objeví původní nezpracovaný obraz, následně zde lze ověřit správnost a vhodnost nastavení na jehož základě bude obraz přetvořen do podoby binárního obrazu, viz kapitola 4.4. Po vykonání všech procedur, které slouží k úpravě obrazu a k nalezení míst, která jsou nejvhodnější pro indent, se v grafickém výstupu zobrazí i vybrané lokace.

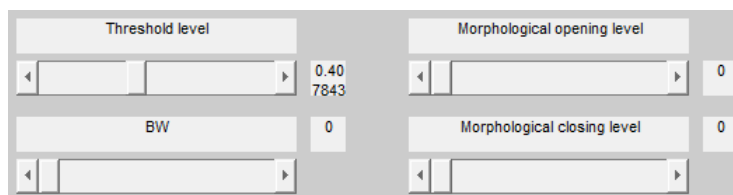
Součástí sekce s grafickým výstupem je i panel, na kterém se zobrazuje název zpracovávaných obrazových dat.



Obrázek 5.2: Grafický výstup

5.2 Nastavení zpracování obrazu

K dispozici jsou celkem čtyři posuvníky, viz obrázek 5.3, které jsou označeny jako Threshold level, Morphological opening level, Morphological closing level a BW. Jejich význam je vysvětlen v kapitole 4. Vedle každého posuvníku se nachází pole vypisující velikost aktuálně nastavené hodnoty.



Obrázek 5.3: Nastavení zpracování obrazu

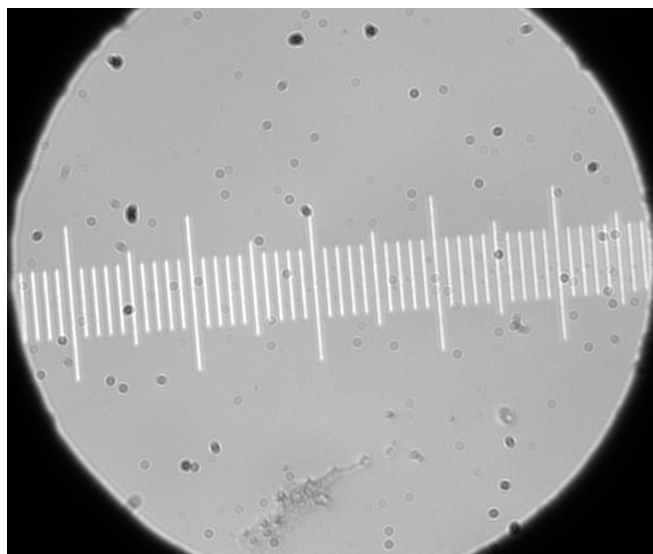
5.3 Uživatelské menu

Uživatelské menu, které slouží k obsluze programu, nabízí sedm základních funkcí. Princip jejich fungování bude popsán v následujícím textu.

5.3.1 Calibration

Funkce, které je v rámci programu označovaná jako Calibration, slouží k výpočtu skutečné velikosti 1 pixelu. Vstupem pro funkci je snímek měrky, viz obrázek 5.4, který je pořízen za stejných podmínek, za kterých budou pořizována i ostatní obrazová data.

Po spuštění funkce je uživatel vyzván k stanovení reálné velikosti měrky. Následně uživatel pětkrát po sobě označí tuto měrku v načteném obrazu. Z pěti hodnot je pak vypočtena průměrná hodnota, která udává velikost jednoho pixelu v mikrometrech. Důvodem proč je prováděno pět měření, je dosažení co možná nejpřesnějšího výsledku.



Obrázek 5.4: Obraz s měrkou

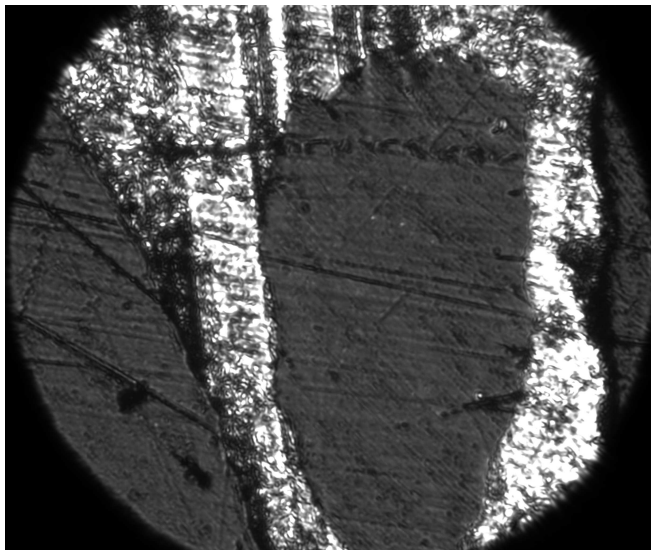
Vypočtený výsledek je zobrazen pod částí programu, které je označeno jako grafický výstup, viz kapitola 5.1.

5.3.2 Load image

V rámci této funkce jsou načtena obrazová data. V případě, že jsou tato obrazová data barevná, dojde k jejich převedení do obrazu v odstínech šedi. Rovněž je v této fázi s využitím Otsuovy metody vypočtena nejvhodnější hodnota thresholdu, viz kapitola 4 a je předběžně přednastaveno na posuvnicích, které ovlivňují jakým způsobem bude šedý obraz převeden do formy binárního obrazu. Navzdory tomuto přednastavení doporučené hodnoty thresholdu má uživatel stále možnost si zvolit takovou hodnotu, kterou uzná za vhodnou.

Výstupem této funkce je pracovní obraz 5.6, který je složen z předem definovaného množství

obrazu, viz obrázek 5.5. Důvodem proč nelze pořídit rovnou pracovní obraz, ale je nutné ho skládat z dílčích obrazů je, že pouze při snímání s velkým zvětšením jsou získána obrazová data v kvalitě použitelné pro vyhodnocení tvrdosti. Z každého dílčího obrazu je vyříznuta část, která je pak využita pro vytvoření pracovního obrazu. Vzdálenost mezi středy dílčích obrazů je taková, aby mezi sebou obrazy, které jsou z nich vyříznuty, co nejplynuleji navazovaly.



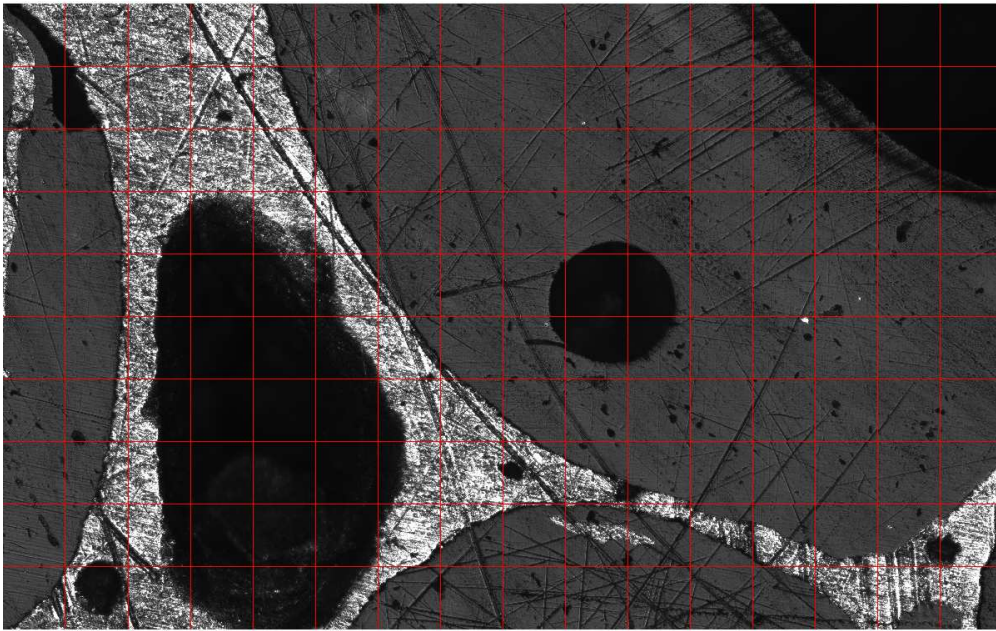
Obrázek 5.5: Dílčí obraz

Před tím než je zahájeno skládání obrazu je uživatel vyzván k zadání počtu obrazů, které budou v řádcích pracovního obrazu a počtu obrazů ve sloupcích. Uživatel rovněž zadává vzdálenost mezi středy dílčích obrazů. Tato vzdálenost pak slouží jako podklad pro vytváření výřezů.

Uživateli je rovněž ukázán náhled z několika dílčích obrazů, čímž je mu dána možnost odříznout z dílčích obrazů část okraje, díky čemuž bude zajištěna lepší návaznost obrazů a pracovní obraz, tak bude přesnější a hodnoty, které z něj budou získány budou bližší skutečnosti.

5.3.3 Show binary image

Tato funkce slouží k ověření vhodnosti výchozího nastavení pro zpracování binárního obrazu. Obraz je zde konvertován do formy binárního obrazu na základě nastavené hodnoty thresholdu a rovněž jsou odstraněny malé objekty. Tato funkce je v programu přítomna proto, aby si uživatel mohl zkontrolovat své nastavení bez toho, aniž by byl nucen spouštět celý program, který je v případě některých procedur pro nalezení nejvhodnějších míst pro indent, relativně časově náročný.



Obrázek 5.6: Pracovní obraz, červeně jsou znázorněny hranice mezi výřezy z dílčích obrazů

5.3.4 Execute

V první fázi po spuštění této funkce jsou vykonány stejné procesy, které byly zmíněny v předchozí podkapitole, jejichž účelem je vytvoření binárního obrazu podle zvolených parametrů. Následně je na základě původního nastavení uživatele zvolena jedna z procedur, které slouží k nalezení nejvhodnějších míst pro indent. K dispozici jsou celkem tři procedury, které jsou označeny jako Teziste, Najdi matici a Nafukovani a princip jejich fungování bude popsán níže. Výstupem funkce Execute je matice o rozměrech $2 \times n$ obsahující souřadnice vybraných lokací pro indent.

Teziste

Tato metoda spočívá v tom, že načtený obraz je rozdělen do několika čtvercových částí. V každé z těchto čtvercových částí je vypočteno, jak velká je její část, která je tvořena materiálem a jak velká je část tvořená póry (v programu je toto reprezentováno jako počet černých a bílých pixelů). Pokud podíl pixelů, které reprezentují materiál, přesahuje jisté procento, je vypočtena poloha těžiště. Výstup procedury je jednak uložen pro další zpracování a jednak je zobrazen v grafickém výstupu tohoto programu.

K výpočtu souřadnic těžiště slouží následující vztah:

$$[x_T, y_T] = \left[\frac{U_y}{S}, \frac{U_x}{S} \right] = \left[\frac{\int_S x dS}{\int_S dS}, \frac{\int_S y dS}{\int_S dS} \right], \quad (5.1)$$

kde x_T a y_T jsou souřadnice těžiště, U jsou statické momenty průřezu, S je plocha průřezu a x a y označují osu souřadnicového systému. Výpočet souřadnic těžišť je řešen pomocí předpřipravených funkcí.

Velkou výhodou této metody je její rychlost, protože na rozdíl od ostatních metod nejsou procházeny všechny pixely. Další výhodou je, že díky tomu, že se zpracovává podíl vhodných pixelů v obraze, mohou být jako místa vhodná pro indent označeny i pozice, které by byly ostatními metodami zamítnuty, protože ty v obraze pátrají po výskytu konkrétní matice.

Najdi matici

Tato metoda je postavena na principu, že ve zpracovávaných obrazových datech pátráme po výskytech matice, která je tvořena body, jenž mají velikost jedna. To znamená, že hledáme takové části obrazu, v kterých se nachází pouze materiál a žádné póry. Po nalezení takovýchto matic je zaznamenána poloha jejich středu. Nalezené body, které splňují kritéria stanovená programem, jsou jednak uloženy pro další zpracování a jednak jsou zobrazeny v grafickém výstupu programu.

Pro řešení problému s hledáním matice ve větší matici, která představuje zpracovávaný obraz, byl použit program `findsubmat` ze služby File Exchange provozované společností MathWorks [33].

Na rozdíl od předchozí procedury musela být tato metoda rozšířena o podmínku, které odstraňuje ty nové body, které se nacházejí příliš blízko již určených poloh pro indent. Nevýhodou této metody oproti té, která byla zmíněna výše, je její velká výpočetní a časová náročnost. Výhodou je, že pokud program vybere nějaký bod, pak máme jistotu, že v jeho okolí bude dostatek prostoru pro provedení indentace. Tuto jistotu u předchozí procedury nemáme, protože v té se zpracovávají jednotlivé podčásti obrazu samostatně.

Nafukovani

Tato procedura spočívá v přejíždění obrazu maskou, tedy hledání takové části obrazu, které je tvořena pouze materiálem a žádnými póry. V případě, že je takové místo nalezeno, je maska zvětšena, nafouknuta. Pokud se podaří dosáhnou určité velikosti masky, která je nastavena uvnitř programu, bylo nalezeno místo vhodné pro indent a lze pokračovat v procházení obrazu. Nalezené body, které splňují kritéria stanovená programem, jsou stejně jako v případě předchozích

procedur jednak uloženy pro další zpracování a jednak jsou zobrazeny v grafickém výstupu programu.

Stejně jako v případě procedury „Najdi matici“ bylo i v tomto případě nutné doplnit program o podmínku, která zajistí, aby se mezi výsledné body vhodné pro provádění indentace nezapisovaly takové body, které od nich nejsou dostatečně vzdáleny.

Výhoda této procedury je stejná jako v předchozím případě, pokud nalezneme nějaký bod, tak v jeho okolí bude skutečně dostatek místa pro provedení indentace. Nevýhodou této procedury je opět její značná výpočetní a časová náročnost, která je ještě výraznější než v předchozím případě, a proto se tato procedura jeví pro praktické použití jako nevhodná.

5.3.5 Add point

Funkce Add point slouží k přidání dalšího bodu pro indent v případě, kdy uživatel není spokojen s počtem navržených bodů. Přidání probíhá přímo kliknutím na požadovanou lokaci.

5.3.6 Delete point

Funkce Delete point slouží k odebrání navržené lokace pro indent v případě, kdy uživatel požaduje počet navržených bodů za zbytečně vysoký, anebo se navržené lokace jeví jako nevhodná. Odebrání probíhá přímo kliknutím do blízkosti bodu, který chce uživatel odebrat. Pokud se v určitém okruhu od místa kliknutí nachází nějaký bod, dojde k jeho smazání. Pokud je v oblasti více bodů, dojde ke smazání toho z nich, který je nejbliže k místu, které zvolil uživatel.

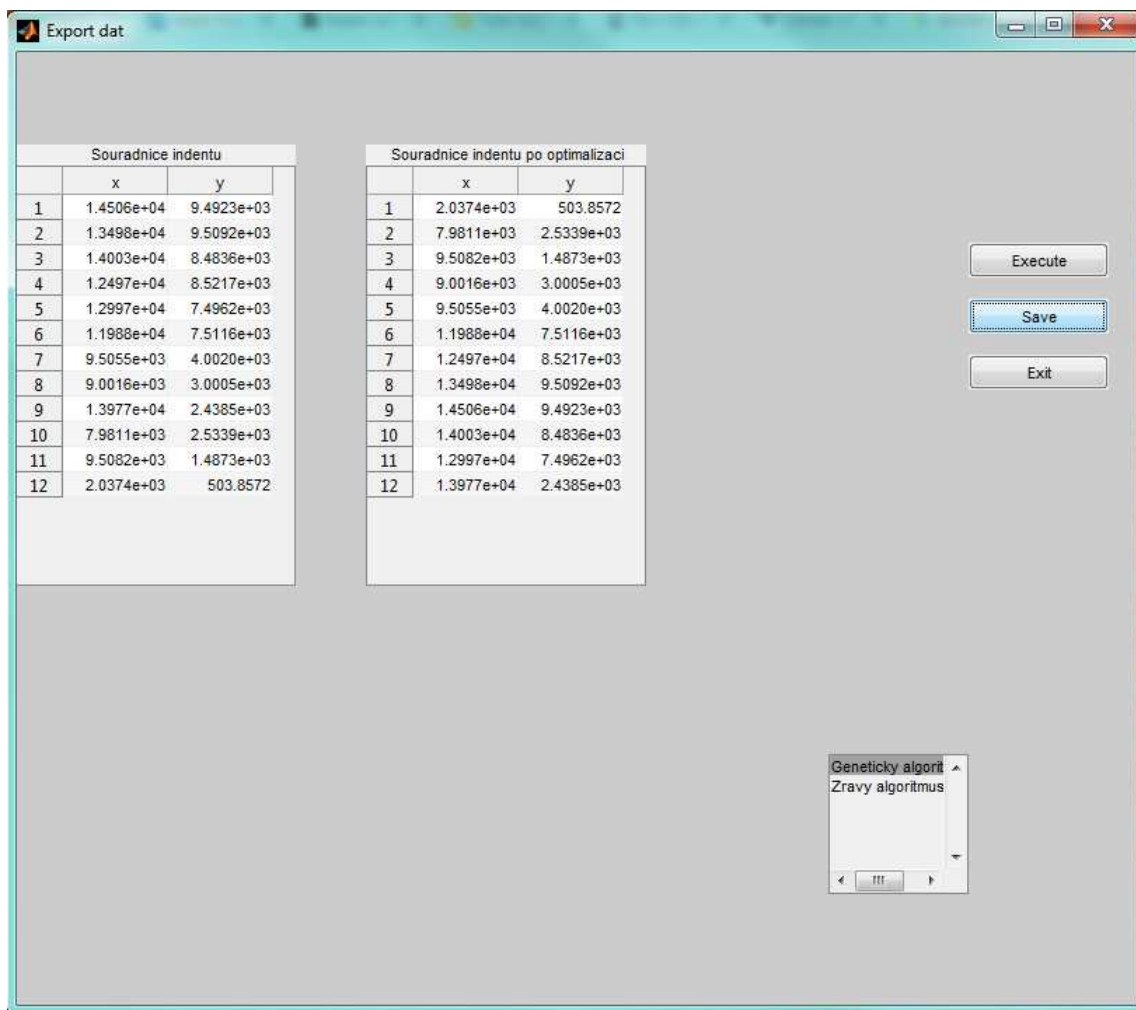
5.3.7 Export

Funkce export otevře uživatel nové okno, viz obrázek 5.7, ve kterém jsou vypsány stanovené souřadnice pro provádění indentace. Uživatel zde má možnost pomocí jedné z předpřipravených funkcí změnit pořadí bodů a tím tak urychlit provádění experimentu (příkaz Execute, protože body budou seřazeny takovým způsobem, aby jejich projití zabralo nejmenší potřebný čas, viz podkapitola 5.4.

Upravené, nebo neupravené finální lokace pro indent lze následně uložit (příkaz Save). Souřadnice jsou uloženy ve formě tak zvaného G-kódu, takže jsou připravené pro následující použití jako vstup pro přístroj provádějící zkoušky tvrdosti.

5.3.8 Exit

Funkce Exit slouží k ukončení činnosti programu.



Obrázek 5.7: Grafické uživatelské rozhraní - funkce Export

5.4 Hledání nejkratší cesty

Problém hledání takové nejkratší cesty, která spojuje všechny body v oblasti s tím, že každým z nich prochází právě jednou je úlohou, která spadá do oblasti teorie grafů. Tam se tento problém označuje jako Problém obchodního cestujícího, což je relativně obtížný problém, který svou komplexností spadá do oblasti kombinatorických úloh diskrétní optimalizace [34].

Pro řešení Problému obchodního cestujícího se používají metody dvojího druhu, jedná se jednak o metody exaktní a jednak o metody heuristické [34]. Mezi exaktní metody patří metody lineárního celočíselného programování, metody hrubé síly a metody typu branch and bound. Nevýhodou těchto metod je, že je vzhledem ke značné výpočetní náročnosti lze použít pouze pro omezený počet bodů. Výhodou druhé skupiny metod, tedy mezi metod heuristických, je jejich rychlost i pro relativně obsáhle soubory dat [34].

V této diplomové práci byly pro vyřešení tohoto přístupu použity dva algoritmy, které jsou popsány v následujících podkapitolách.

5.4.1 Hamiltonovská kružnice

Drtivá většina přístupů pro řešení Problému obchodního cestujícího vychází z teorie Hamiltonovských kružnic [34]. Hamiltonovská kružnice je takový podgraf zkoumaného grafu, který je kružnicí a zároveň obsahuje všechny vrcholy grafu. Nezbytnými podmínkami pro to, aby nějaký graf obsahoval Hamiltonovskou kružnici, jsou následující:

- graf musí být souvislý,
- graf nesmí obsahovat hrany typu most,
- graf nesmí obsahovat artikulace,
- graf nesmí obsahovat visící vrcholy [34].

První podmínka nám říká, že pro každé dva vrcholy x, y existuje sled z x do y [35]. Podle druhé podmínky nesmí v grafu existovat taková hrana, po jejímž odebrání by vzrostl počet komponent grafu [34]. Podle třetí podmínky nesmí v grafu existovat takový vrchol, po jehož odebrání se graf rozpadne na více podgrafů [36]. Závěrečná, čtvrtá, podmínka nám říká, že v grafu nesmí existovat taková hrana, jejímž odebráním odebereme z grafu právě jeden vrchol [34]. Vzhledem k povaze řešené úlohy lze všechny zmíněné podmínky považovat za splněné.

5.4.2 Genetický algoritmus

Jádro této procedury bylo převzato ze služby File Exchange provozované společností MathWorks [37], kde ji lze nalézt pod označením Open Traveling Salesman Problem - Genetic Algorithm. Efektivita této převzaté procedury bude porovnána s procedurou vytvořenou autorem.

5.4.3 Hladový algoritmus

Hladový algoritmus slouží k určování minimální Hamiltonovské kružnice, viz kapitola 5.4.1. Bohužel vzhledem k heuristice není zaručeno, že výsledná nalezená kružnice bude skutečně minimální. Hladový algoritmus lze zjednodušeně popsat následujícím způsobem:

- 1. krok

- Ze zadaných vrcholů vybereme počáteční vrchol
- zařazené počátečního vrcholu do Hamiltonovské kružnice
- 2. krok
 - Ze všech hran, které vycházejí z počátečního vrcholu, vybereme tu s nejmenším ohodnocením
 - Vrchol, v kterém tato hrana končí, zařadíme do Hamiltonovské kružnice
- 3. krok
 - vrchol, který jsme získali v předchozím kroku, si zvolíme za nový počáteční vrchol
 - pokud už nám žádné další vrcholy nezbývají pokračujeme krokem 4, jinak se vracíme do kroku 2
- 4. krok
 - Získali jsme hledanou hamiltonovskou kružnici [34]

5.5 Dupočítání souřadnic

Po vykonání postupů popsaných v předchozích kapitolách získáme polohu budoucích indentů, ale zatím ji známe pouze v rámci pracovního obrazu. V další fázi je nutné tyto pozice přepočítat na reálné souřadnice, se kterými pracuje instrumentovaný indenter.

K přepočítání na reálné souřadnice je potřeba jednak znát reálnou velikost jednoho pixelu a jednak reálnou polohu jednoho místa v obraze. Postup výpočtu velikosti jednoho pixelu je popsán výše. Jako bod jehož reálnou polohu známe nám slouží první pořízen snímek, který se v pracovním obrazu nachází vpravo dole. Střed tohoto dílčího obrazu je identický se startovní pozicí stroje.

Převedení na reálné souřadnice pak probíhá tak, že nejdříve změním souřadnicový systém takovým způsobem, aby souřadnice $[0,0]$ byla v pravém dolním rohu obrazu. To uděláme tak že od délky a šířky obrazu odečteme souřadnici x respektive y indentu. Všechny nově získané souřadnice vynásobíme skutečnou velikostí jednoho pixelu. Posledním krokem je započítání posunu, kterým se odlišuje tento souřadnicový systém od souřadnicového systému, v kterém pracuje indenter. K výpočtu tohoto posunu nám slouží znalost skutečné pozice středu prvního dílčího obrázku.

5.6 Výstup

Výstupem softwarové části této diplomové práce je G-kód pro počítačem číslicově řízenou (CNC) experimentální proceduru. Tento kód je vytvořen v takové podobě, aby experimentální přístroj urazil při procházení všech určených míst pro indent co nejkratší možnou cestu. Optimalizace této trasy je řešena jedním z algoritmů, které byly popsány v kapitole 5.4. Samotný G-kód je podrobněji popsán v kapitole 6.3.

Kapitola 6

Experiment

V rámci této kapitoly bude popsán postup, kterým bylo splněno jedno z dílčích zadání této diplomové práce, tedy vytvoření zdrojového kódu (G-kódu) pro počítačem číslicově řízenou (CNC) experimentální proceduru. V této kapitole je rovněž ve stručnosti představen stroj, pro který je celá procedura navrhována. Funkčnost této procedury bude ověřena v kapitole 7.

6.1 Automatizovaný přístroj - instrumentovaný indentor

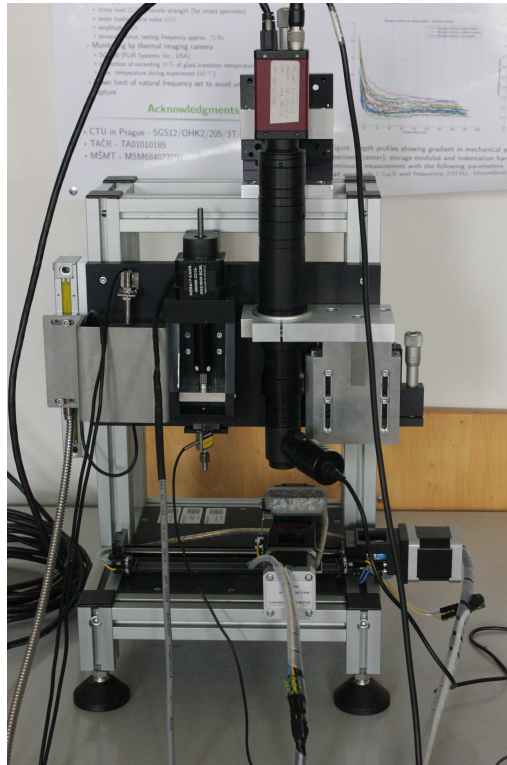
Automatická procedura pro mapování povrchové tvrdosti porézních materiálů na základě indentačních zkoušek je napsána pro mikroindentační zařízení, viz obrázek 6.1. Toto zařízení bylo vytvořeno na Ústavu mechaniky a materiálů Fakulty dopravní ČVUT v Praze v rámci jednoho z řešených grantů. Zařízení, které je v této kapitole popsáno, vzniklo modifikací konstrukce popsané v [38].

Zařízení bylo sestaveno kvůli potřebě automatizovat měření tvrdosti na ploše s vytvořením mřížky vtisků pro tvorbu map tvrdosti. Velikost mřížky vtisku je limitována funkčním rozsahem zařízení. Výhodou automatizovaného zařízení je výrazné snížení chybovosti, protože takovéto zařízení není zatíženo lidskou chybou [38].

Zařízení je koncipováno na provádění zkoušky tvrdosti podle Vickerse, ale je možná výměna hrotu a provádět tak jiný typ zkoušky, viz kapitola 3. Zařízení je vybaveno kamerou a přesným polohováním vzorku, a tak lze provádět indentaci na konkrétním zvoleném místě, anebo provádět celou sérii měření [38].

Indentor je aktuálně osazen siloměrem LSB200 (FUTEK, USA) s měřícím rozsahem 8,90 N.

Instrumentovaný indentor je schopen provádět zkoušky tvrdosti pod nízkým zatížením. Zařízení obsahuje tři motorizované osy. Dvě z nich slouží k polohování vzorků. Poslední z nich je



Obrázek 6.1: Instrumentovaný indentor

určena k provádění zkoušky tvrdosti. Zařízení je dále vybaveno CCD kamerou (Manta G-504B) s rozlišením 2452×2056 px. Účelem CCD kamery je jednak pořízení obrazových dat, která slouží jako podklad pro určení lokací pro indentaci, a jednak vyhodnocení tvrdosti.

K řízení zkoušky slouží LinuxCNC, což je software, který byl navržen pro počítačové řízení CNC zařízení. Konkrétně byla použita verze 2.6.4. Výhodou tohoto řídicího prostředí je jeho bezplatná distribuce [39], [40].

Samotné zařízení lze rozdělit na dvě části. Jedná se o následující části - mechanickou a optickou.

6.1.1 Mechanická část

Kostru celého zařízení tvoří hliníková slitina s průřezem 30×30 mm. Pro polohování vzorku je zařízení vybaveno dvěma motorizovanými polohovacími osami s přesností $10 \mu\text{m}$. Pro potřeby provádění zkoušek tvrdosti je zařízení vybaveno indentační osou s přesností $1,5 \mu\text{m}$, která je zakončena dvojicí koncových spínačů. Na indentační ose je umístěna příruba pro připevnění indentoru a siloměru [38].

6.1.2 Optická část

Optická část instrumentovaného indentoru se skládá z CCD kamery (Manta G-504B, Německo), objektivu (Shenzhen Hayear Electronics Co. Ltd., China) a z koaxiálního osvětlení, které bylo použito za účelem zvýšení kvality obrazových dat. Optická soustava je připevněna k lineárnímu stolku, který disponuje mikrometrickým šroubem. Účelem tohoto šroubu je, pomocí přibližování a oddalování od povrchu vzorku, ostřit obraz.

6.1.3 Enkodér

Enkodér je součást zařízení, která umožňuje vyčítání polohy své pohyblivé části a předání této hodnoty řídicímu systému. Této informace o poloze lze využít pro řízení krokových motorů, kdy je poloha určená enkodérem porovnávána s polohou, kterou udává instrukce pro krokový motor [41]. Na instrumentovaném indentoru je enkodér použit pouze pro vyčítání závislosti síly na indentační hloubce.

6.2 Modulární jednotka

Pro řízení experimentu byla použita modulární řídicí jednotka, viz obrázek 6.2. Tato jednotka vznikla v rámci jedné z diplomových prací, které byly v minulosti zadány ústavem K618 [42]. Řídicí jednotka byla sestavena za účelem poskytnout otevřené řešení pro řízení experimentálních zařízení, která jsou využívána v rámci vědeckovýzkumné činnosti ústavu Mechaniky a materiálů.

Řídicí jednotka je složena z několika základních částí, konkrétně se jednotka skládá z řídicího počítače, elektroniky pro řízení motorů a ostatní podpůrné elektroniky, viz obrázek 6.3 [42].

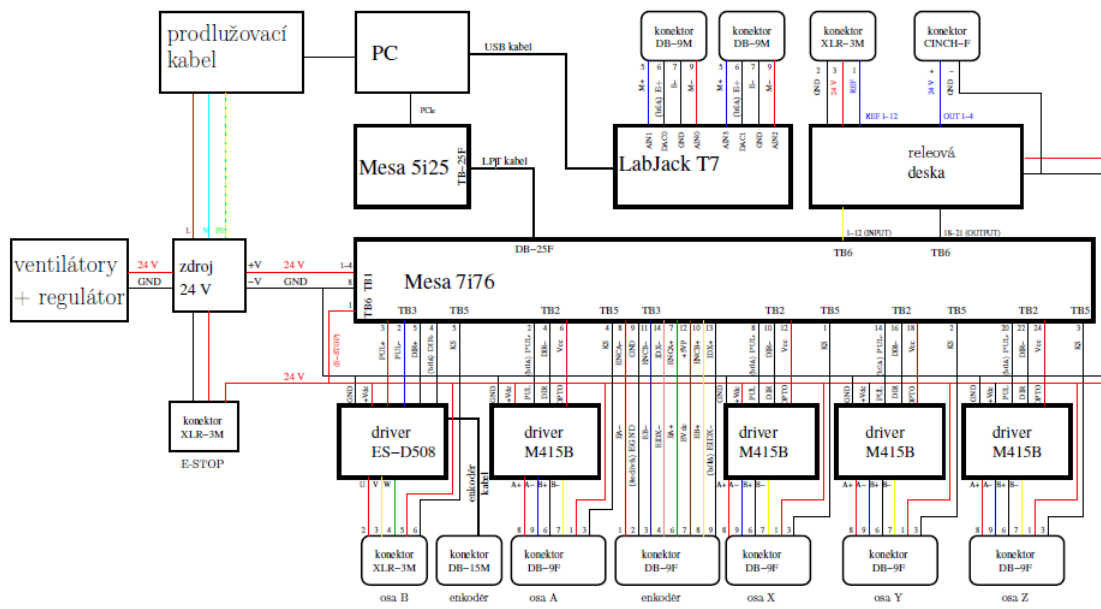
6.2.1 Vlastnosti řídicí jednotky

V této kapitole budou shrnuty vybrané schopnosti řídicí jednotky:

- řízení tří motorizovaných os, které jsou osazeny krokovými motory s koncovými spínači,
- řízení jedné motorizované osy, které je osazena krokovým motorem s koncovými spínači a enkodérem,
- řízení jedné motorizované osy, které je osazena servo-motorem s koncovými spínači a enkodérem,
- vyčítání dvou signálů ze siloměrů a tenzometrů,



Obrázek 6.2: Modulární jednotka [42]



Obrázek 6.3: Modulární jednotka - schéma [42]

- ovládání fotoaparátu a kamery [42]

6.3 G-kód

Jako výstup této diplomové práce byl v zadání navrhnout G-kód. Tento programovací jazyk slouží k řízení CNC obráběcích strojů. Historie tohoto programovacího jazyka sahá do šedesátých let, z důvodu relativně velkého stáří tohoto jazyka se většina výrobců nedrží předdefinovaných standardů a do jazyka přidává různé změny [43]. Kód umožňuje přikazovat stroji kam a jakým způsobem má přesunovat pracovní nástroj.

G-kód získal své označení z toho důvodu, že většina příkazů začíná písmenem G, ale ve skutečnosti se v tomto kódu vyskytují příkazy, které používají všechna písmena abecedy. Typický příkaz v G-kódu je tvořen písmenem abecedy, které je následované číslem [44]. V kódu, který je generován v této diplomové práci, jsou použity i funkce začínající písmenem M. Jedná se o dodatečně vytvořené funkce.

6.3.1 Vysvětlení kódu

Pro seznámení se s principy fungování G-kódu byly použity následující zdroje: [45], [46], [47]. Ukázka jak vypadá výstup tohoto programu je k nahlédnutí v příloze B. V této kapitole bude popsán význam jednotlivých řádků tohoto kódu. Pro vysvětlení fungování G-kódu je v této kapitole vložena vzorová struktura G-kódu, viz obrázek 6.1.

Zdrojový kód je zahájen řádkem začínající písmenem O, tento řádek slouží jako identifikátor celé procedury. Po názvu procedury následují tři příkazy - M125, M123 a G90. První z nich zajistí, že dokument logfile bude prázdný, druhý z nich slouží k zahájení vykreslování grafu a poslední z příkazů řekne stroji, že vstupem pro jeho činnost budou souřadnice indentů a nikoliv vzdálenosti mezi nimi.

Mezi závorkami (indentation) a (indentation end) se nachází část kódu, který řídí indentaci. První řádek přikazuje stroji, aby se maximální rychlostí přesunul na zadané souřadnice. K souřadnicím jsou přidány příkazy, které začínají symbolem #. Tyto příkazy řeší rozdíl v poloze mezi kamerou a indentorem. Po přesunu na správnou pozici je zavolána funkce *O<indentation_ instrumented>*, která řeší samotné provedení indentace. Tato sada instrukcí je zakončena příkazem G90, který zajistí, že je zachován pohyb v absolutním režimu. Tato sekvence kódu je zkopírována tolikrát, kolik je třeba projít lokací.

Další sada instrukcí je ohraničena závorkami (image) a (image end). V ní je kromě instrukcí,

```

O<automated_positioning> sub
M125 (clear log file)
M123 (live plot)
G90 (absolute mode)

(indentation)
G0 X[xx+#5168] Y[yy+#5169]
O<indentation_instrumented> call
G90
(indentation end)

M124 (export log file)

G90
(image)
G0 Xxx Yyy
M126 (make picture)
(image end)

M2 (end of program)

```

Obrázek 6.4: Vzorová struktura G-kódu

kteří slouží k pohybu mezi lokacemi, umístěn i příkaz M126, jehož úkolem je pořizovat snímky míst, kde byla provedena indentace. Tato sekvence kódu je opět zkopírována tolikrát, kolik je třeba projít lokací.

G-kód je zakončen příkazem M2, který ukončí jeho činnost.

6.4 Materiál

Testovaným materiálem byla kovová pěna, které se v průmyslové praxi označuje jako ALCORAS® (AlCarbon, Německo). Tato kovové pěna je tvořena z 97 % hliníkem, z 1,5 % vápníkem a z 1,5 % titanem. Pěna má póry o velikosti 2 – 4 mm, které jsou od sebe odděleny stěnami o tloušťce

100 – 2000 μm [39].

6.5 Příprava vzorku

Zkoumaný materiál byl nařezán na menší kvádry s minimální tloušťkou 12 mm. Tím byla zajištěna integrita vzorku. K řezání vzorku byl použita řezná kapalina (Isomet 1000, Buehler GmbH, Německo). Řezání vzorků bylo prováděno za nízkých rychlostí (3 mm/min, s otáčkami 800 rpm), aby tak byla minimalizována veškerá povrchová poškození, která by mohla ovlivnit výsledky zkoušky tvrdosti. Následně byl připravený a očištěný vzorek uložen do zalévací hmoty (VeriK-leer, Buehler GmbH, Německo). Poté došlo k jeho vybroušení a leštění. K leštění a broušení byly použity kotouče vyráběné z karbidu křemíku. Cílem těchto procedur bylo jednak odstranění povrchové vrstvy, která mohla být narušena předchozí manipulací s materiálem a jednak získat materiál s minimální hrubostí povrchu, který je vhodný pro provádění zkoušek tvrdosti [39]. Povrch materiálu musí být rovný, aby nedošlo k deformaci indentů a tím ke zkreslení výsledků měření.

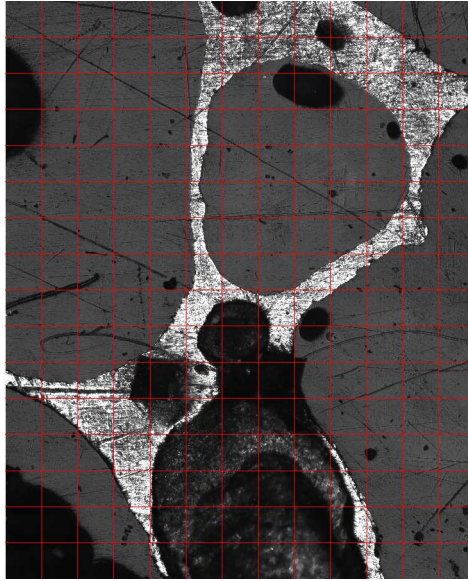
6.6 Popis kalibrace

Pro ověření přesnosti celého zařízení byly vytvořeny dvě sady indentů. První sadou byla čtvercová mřížka tvořená devíti indenty. Velikost strany tohoto čtverce byla 0,75 mm. Druhá sada indentů byla tvořena řadou čtyř indentů. Vzdálenost mezi krajními indenty činila 1 mm. Cílem kalibrace bylo zjistit, jak přesně instrumentovaný indenter polohuje. Předem známá vzdálenost mezi indenty byla opticky ověřena.

6.7 Popis experimentu

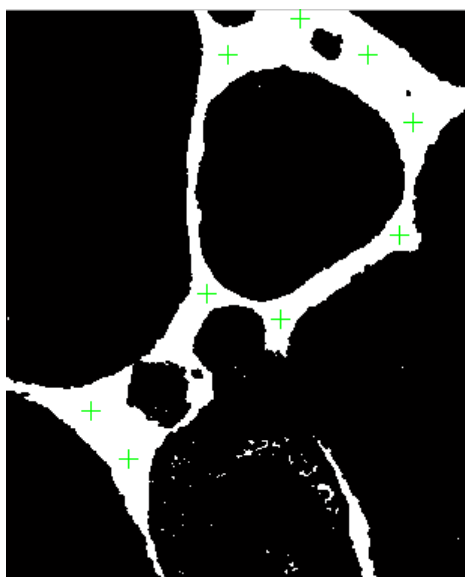
Funkčnost vytvořeného programu byla ověřena provedením měření tvrdosti podle Vickerse. Tvrdost podle Vickerse se stanoví použitím vztahu 3.3.

V první fázi experimentu byly pořízeny snímky zkoumaného vzorku. Tyto snímky byly složeny do jednoho pracovního obrazu, viz obrázek 6.5. Postup skládání obrazu je popsán v kapitole 5.3.2. Důvodem pro vytváření pracovního obrazu z několika stovek dílčích snímků je požadavek na dostatečné rozlišení. Pracovní obraz byl následně zpracován pomocí funkce „Tezisté“, která byla rovněž popsána v kapitole 5.3.4. Cílem použití těchto funkcí byl výběr vhodných lokací pro indent, viz obrázek 6.6.

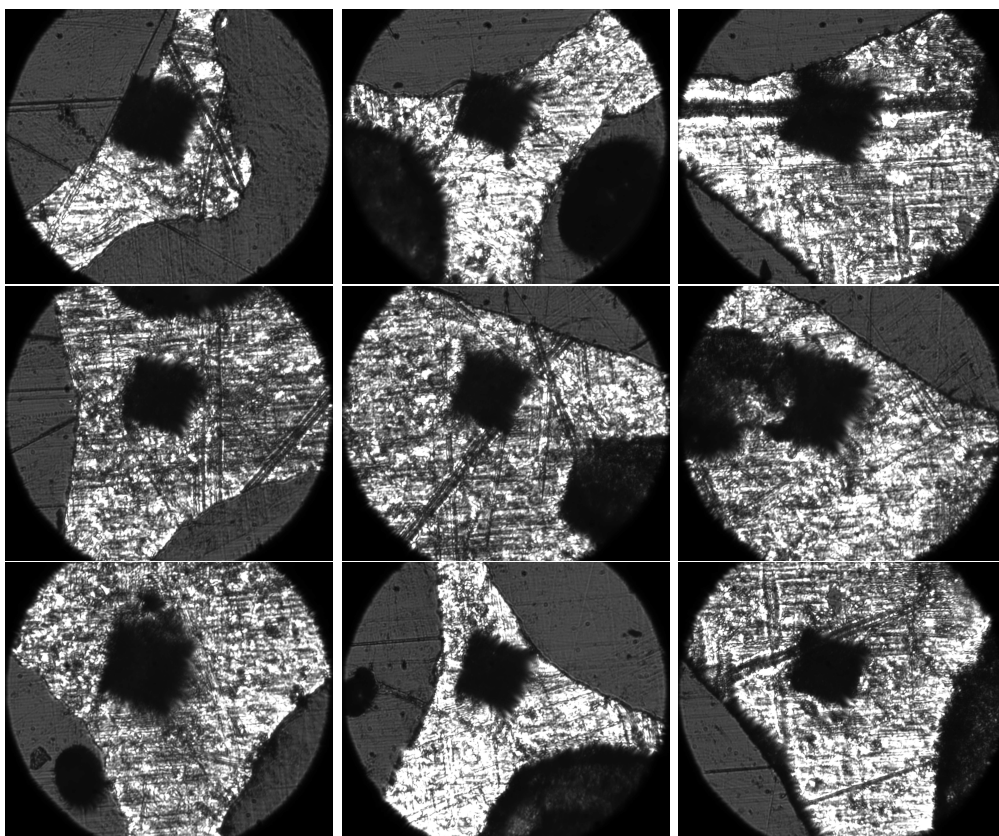


Obrázek 6.5: Pracovní obraz (reálná velikost zobrazené plochy je $2,84 \times 3,49$ mm)

Takto získané lokace byly pomocí genetického algoritmu optimalizovány takovým způsobem, aby jejich projití zabralo nejkratší možný čas. Následně byl automaticky vygenerován G-kód, který instruoval přístroj, jakým způsobem má procházet vzorek. Ve všech určených souřadnicích byla provedena indentace s použitím síly 3 N. Následně byly tyto souřadnice projity podruhé a byly pořízeny snímky indentů, viz obrázek 6.7, které sloužily jako podklad pro vyhodnocení experimentu.



Obrázek 6.6: Pracovní obraz s identifikovanými lokacemi pro indentaci (reálná velikost zobrazené plochy je $2,84 \times 3,49$ mm)



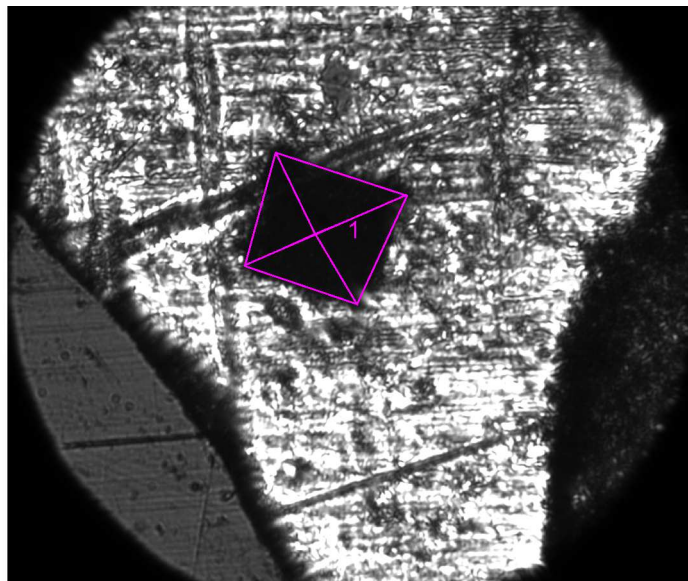
Obrázek 6.7: Snímky indentů

Kapitola 7

Výsledky

7.1 Vyhodnocení experimentu

Provedený experiment byl vyhodnocen pomocí skriptu, který vznikl v rámci jedné z bakalářských prací zadávaných ústavem K618 [17], výsledek aplikace skriptu viz obrázek 7.1.



Obrázek 7.1: Příklad aplikace skriptu pro měření velikosti tvrdosti

Postupně byly načteny snímky jednotlivých indentů. Na nich byl manuálně označen střed a vrcholy indentů. Z určených hodnot, z velikosti působící síly a z dopočítané velikosti jednoho pixelu, byla vypočtena velikost tvrdosti podle Vickerse HV na základě vzorce 3.3, viz tabulka 7.1.

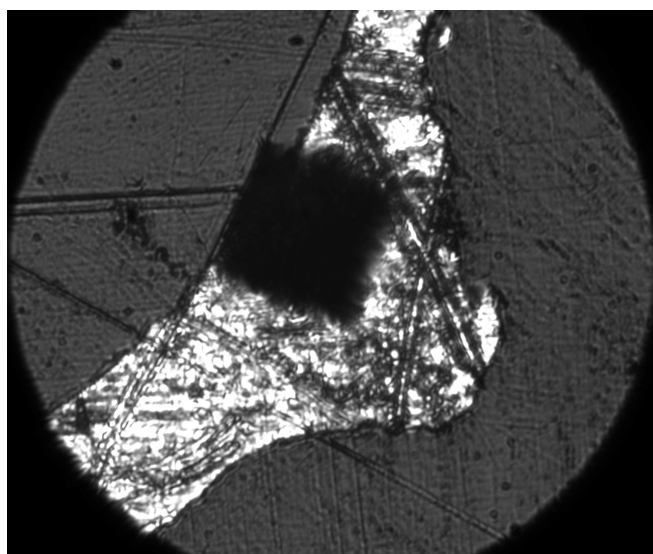
Z naměřených velikosti tvrdosti byla spočtena průměrná hodnota a velikost směrodatné odchylky:

u_1 [mm]	u_2 [mm]	d [mm]	HV
0,14425	0,14599	0,14512	26,936
0,13815	0,14798	0,14307	27,715
0,14312	0,14205	0,14259	27,904
0,14627	0,14345	0,14486	27,036
0,14205	0,14780	0,14493	27,010

Tabulka 7.1: Tabulka naměřených a vypočtených hodnot

$$27,3202 \pm 0,2053HV0,3059. \quad (7.1)$$

Z nalezených lokací bylo nutné některé z nich vyloučit. Důvodem pro vyloučení bylo, že některé z uskutečněných indentů byly v nedostatečné vzdálenosti od okraje materiálu, viz obrázek 7.2.



Obrázek 7.2: Příklad špatně provedené indentace

7.2 Hodnocení chyby polohování

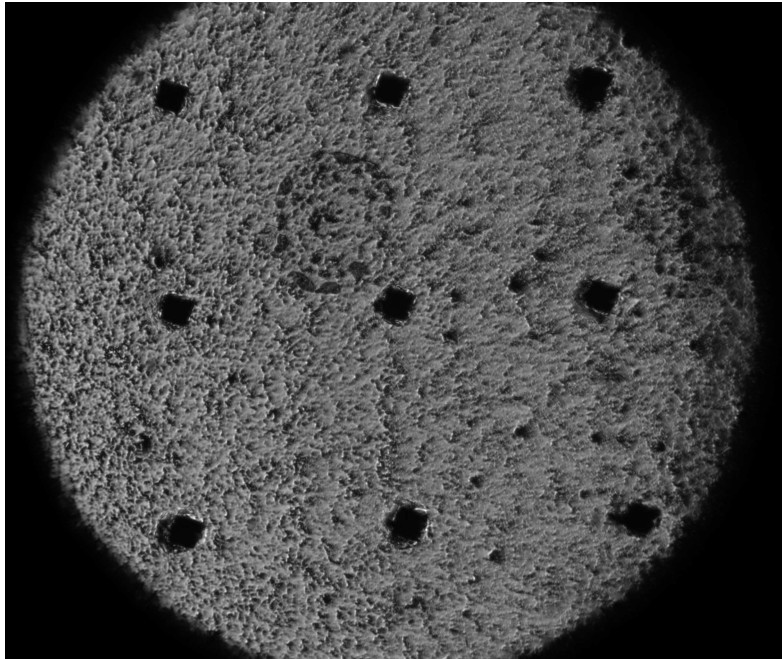
K vypočtení chyby polohování byly provedeny dvě série indentací. V první z nich bylo provedeno devět vpichů. Tyto indenty jsou uspořádány do čtvercové mřížky, velikost hrany této mřížky je

0,75 mm, viz obrázek 7.3. V druhé sérii indentací byla vytvořena řada čtyř indentů o celkové délce 1 mm, viz obrázek 7.4.

Chyba zařízení byla vypočtena z následujícího vztahu:

$$chyba = \frac{(l - l_{str})100}{l}, \quad (7.2)$$

kde l_{str} je aritmetický průměr naměřených hodnot a l je předpokládaná velikost měřeného úseku.



Obrázek 7.3: Indenty ve čtvercové mřížce

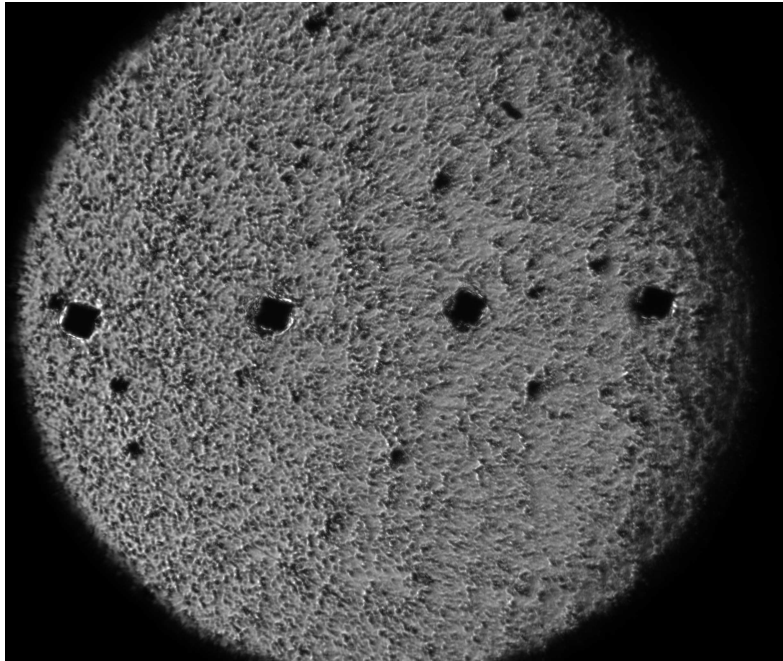
7.2.1 Indenty ve čtvercové mřížce

Nejdříve byly přeměřeny vzdálenosti mezi krajními indenty, viz tabulka 7.2. Vzdálenost mezi nimi by měla být 0,75 mm. Dosazením do rovnice 7.2 bylo vypočteno:

l_1 [mm]	l_2 [mm]	l_3 [mm]	l_4 [mm]	l_5 [mm]	l_6 [mm]
0,7255	0,7271	0,7805	0,7460	0,7415	0,7406

Tabulka 7.2: Měření délky o předpokládané velikosti 0,75 mm

$$chyba = 0,86222 \%. \quad (7.3)$$



Obrázek 7.4: Indenty v řadě

Po té byly změřeny vzdálenosti mezi sousedními indenty, viz tabulka 7.3. Tato vzdálenost by měla být 0,375 mm.

l_1 [mm]	l_2 [mm]	l_3 [mm]	l_4 [mm]	l_5 [mm]	l_6 [mm]
0,3759	0,3409	0,3693	0,3542	0,3805	0,3933
l_7 [mm]	l_8 [mm]	l_9 [mm]	l_{10} [mm]	l_{11} [mm]	l_{12} [mm]
0,3628	0,3782	0,3671	0,3722	0,3567	0,3876

Tabulka 7.3: Měření délky o předpokládané velikosti 0,375 mm

Dosazením rovnice 7.2 bylo vypočteno:

$$chyba = 1,36222 \%. \quad (7.4)$$

Chybovost pro vzdálenosti indentů ve čtvercové mřížce se pohybuje pod hodnotou 1,5 %.

7.2.2 Indenty v řadě

U indentů v řadě byly přeměřena vzdálenost mezi krajními indenty, viz tabulka 7.4. Tato vzdálenost by měla odpovídat vzdálenosti 1 mm

l_1 [mm]
0,9860

Tabulka 7.4: Měření délky o předpokládané velikosti 1 mm

Dosazením do vztahu pro chybovost 7.2 bylo vypočteno:

$$chyba = 1,4 \%. \quad (7.5)$$

Dále byly přeměřeny vzdálenosti mezi indenty, které by měly odpovídat 0,66 mm, viz tabulka 7.5.

l_1 [mm]	l_2 [mm]
0,6623	0,6536

Tabulka 7.5: Měření délky o předpokládané velikosti 0,66 mm

Dosazením do vztahu pro chybovost 7.2 bylo vypočteno:

$$chyba = 0,31 \%. \quad (7.6)$$

Následovalo měření vzdáleností, které by měly odpovídat velikosti 0,33 mm 7.6.

l_1 [mm]	l_2 [mm]	l_3 [mm]
0,3325	0,3292	0,3240

Tabulka 7.6: Měření délky o předpokládané velikosti 0,33 mm

Dosazením do vztahu pro chybovost 7.2 bylo vypočteno:

$$chyba = 0,43 \%. \quad (7.7)$$

Hodnoty, které vyšly pro chybovost pro indenty v řadě se rovněž pohybují pod hodnotou 1,5 %.

Kapitola 8

Závěr

Na základě zadání této diplomové práce byl vytvořen softwarový nástroj sloužící k lokalizaci nejvhodnějších míst pro provedení indentace při měření tvrdosti materiálu za účelem získání přesnější představy o vlastnostech zkoumaného materiálu. Automatizovaná procedura pro povrchové měření tvrdosti porézních materiálů byla vyvinuta jednak s ohledem na příslušné normy a jednak s ohledem na časovou náročnost celého procesu. Procedura je vybavena grafickým uživatelským rozhraním (GUI), které uživateli umožňuje snadnou práci. Uživatel má možnost ovlivňovat nastavení zpracování obrazu, volit si jednu ze tří funkcí, která slouží k identifikaci nejvhodnějších míst pro provedení indentace a zvolit způsob jakým bude provedena optimalizace procházení indentů.

8.1 Zhodnocení softwarového nástroje

Ze tří předpřipravených funkcí určených pro lokalizaci nejvhodnějších míst pro provedení indentace se jako nejvhodnější ukázala funkce s označením „Teziste“, která je i pro velké množství dat relativně rychlá. Další navržené funkce - „Najdi matici“ a „Nafukovani“ se neukázaly jako použitelné. Jejich nevýhodou je značná náročnost na potřebný výpočetní stav.

8.2 Zhodnocení experimentu

Funkčnost navrženého nástroje byla ověřena provedením experimentu, v rámci kterého byla měřena velikost tvrdosti podle Vickerse pomocí instrumentovaného indentoru. Indentační zkouška byla prováděna na materiálu s obchodním označením ALCORAS. Výsledná tvrdost činila:

8.3 Zhodnocení přesnosti polohování

Pro zhodnocení přesnosti zařízení byly provedeny dvě sady indentů. V první z nich byla vytvořena čtvercová mřížka tvořena devíti indenty, v druhé pak byla vytvořena řada čtyř indentů. Následně byly přeměřeny vzdálenosti mezi jednotlivými indenty. Během hodnocení přesnosti polohování bylo zjištěno, že chybovost se ve všech případech pohybuje pod hodnotou 1,5% a tedy je dostatečně nízká na to, aby bylo možné použít dílčí výsledky získané aplikací skriptu pro hledání lokací indentů.

Literatura

- [1] Xu, Y., Li, Y., Zhang, A., Bao, J.: *Epoxy foams with tunable acoustic absorption behavior*, 2017
- [2] Ishizaki, K., Komarmeni, S., Nanko, M.: *Porous materials: Process technology and applications*, Kluwer academic publishers, 1998
- [3] Zlámál, P., Fíla, T., Jiroušek, O., Králík, V.: *Investigation of deformation behaviour of an aluminium foam under high-strain rate loading and comparison with the conventional energy absorbing material*, 2014
- [4] Banhart, J.: *Manufacture, characterisation and application of cellular metals and metal foams*, Progress in Materials Science, Volume 46, Issue 6, 2001
- [5] Callister, W. D. jr.: *Fundamentals of Materials Science and Engineering*, John Wiley & Sons, 2001
- [6] Ashby, M. F., Evans, A. G., Fleck, N. A., Gibson, L. J., Hutschinson, J. W., Wadley, H. N. G.: *Metal Foams: A Design Guide*, Butterworth-Heinemann, 2000
- [7] Doktor, T.: *Diplomová práce: Assessment of pore size distribution using image analysis*, České vysoké učení technické v Praze, vedoucí práce Kytýř, D., 2011
- [8] Gibson, L. J., Ashby, M. F.: *Cellular Solids: Structure and Properties*, Cambridge University Press, 1999
- [9] Abel, A. B., Willis, K. L., Lange, D. A.: *Mercury intrusion porosimetry and image analysis of cement-based materials*, Journal of Colloid and Interface Science, 1999
- [10] Olvera, L. I., Ruiz-Treviño, F. A., Balmaseda, J., Ronova, I. A., Zolotukhin, M. G., Carreón-Castro, M. P., Lima, E., Cárdenas, J., Gaviño, R.: *Microporous polymers from superacid*

- catalyzed polymerizations of fluoroketones with p-quaterphenyl: Synthesis, characterization, and gas sorption properties*, Polymer (United Kingdom), 2016
- [11] Sneek, T., Oinonen, H.: *Measurements of pore size distribution of porous materials*, State Institute for Technical Research, Helsinki, 1970
- [12] Wang, G., Yang, H., Jiao, M., Wei, Y., Li, H., Gong, Z.: *Development and application of aluminum foam material in automobiles*, 2015
- [13] Gupta, N., Zeitmann, S. E., Shunmugasamy, V. C., Pinisetty, D.: *Applications of polymer matrix syntactic foams*, 2014
- [14] Koudelka, P., Zlámal, P., Fíla, T.: *On the Modelling of Compressive Response of Closed-Cell Aluminium Foams under High-strain Rate Loading*, 2014
- [15] Tisha, D., Indranil, G.: *Prospective of employing high porosity open-cell metal foams in passive cryogenic radiators for space applications*, 2016
- [16] Khezerloo, A., Owhadi Esfahani, A., Jalily Lng, S.: *Simulation of noise reduction in passenger trains using metal foams*, 2011
- [17] Fenclová, N.: *Bakalářská práce: Poloautomatické mapování tvrdosti povrchu materiálů*, České vysoké učení technické v Praze, vedoucí práce, vedoucí práce Kytýř, D., 2013
- [18] Krčmářová, N., Šleichrt, J., Fíla, T., Koudelka, P., Kytýř, D.: *Deformation Behaviour of Gellan Gum Based Artificial Bone Structures under Simulated Physiological Conditions*, České vysoké učení technické, 2016
- [19] Kytýř, D., Krčmářová, N., Šleichrt, J., Fíla, T., Koudelka, P., Gantar, A., Novak, S.: *Deformation Response of Gellan Gum Based Bone Scaffold Subjected to Uniaxial Quasi-static Loading*, Acta Polytechnica, 2017
- [20] Amini, A. R., Laurencin, C. T., Nukavarapu, S. P.: *Bone tissue engineering: Recent advances and challenges*, Critical Reviews in Biomedical Engineering, 2012
- [21] Ludvík, J., Bilek, K., Ludvík, Š.: *Zkoušky tvrdosti: Přehled základních zkušebních metod pro uživatele tvrdoměrů Qness*, Metrotest Kladno, 2010
- [22] Doležal, P., Pacal, B.: *Hodnocení mikrotvrdosti struktur materiálů*

- [23] Machek, V., Sodomka, J.: *Nauka o materiálu - 2. část: Vlastnosti kovových materiálů*, Nakladatelství ČVUT, 2007
- [24] ČSN EN ISO 6506: *Kovové materiály - Zkouška tvrdosti podle Brinella*, 2015
- [25] ČSN EN ISO 6507: *Kovové materiály - Zkouška tvrdosti podle Vickerse*, 2006
- [26] ČSN EN ISO 6508: *Kovové materiály - Zkouška tvrdosti podle Rockwella*, 2017
- [27] ČSN EN ISO 4545: *Kovové materiály - Zkouška tvrdosti podle Knoopu*, 2006
- [28] Russ, J. C., Neal, F. B.: *The Image Processing Handbook, Seventh Edition*, CRC Press, 2015
- [29] Gonzales, R. C., Woods R. E., Eddins S. L.: *Digital Image processing using MATLAB*, Gatesmark Publishing, 2009
- [30] Shapiro, L. G., Stockman, G. C.: *Computer vision*, Prentice Hall, 2001
- [31] Hos, J.: *Bakalářské práce: Určování orientace vláken v krátkovláknových kompozitech analýzou obrazových dat*, České vysoké učení technické v Praze, vedoucí práce Doktor, T., Kytýř, D., 2015
- [32] Image Processing Toolbox. <https://www.mathworks.com/products/image.html>. [Online; navštíveno 21. 1. 2017]
- [33] findsubmat. https://www.mathworks.com/matlabcentral/fileexchange/23998-findsubmat?s_tid=srchtitle. [Online; navštíveno 16. 2. 2017]
- [34] Volek, J., Linda, B.: *Teorie grafů - aplikace v dopravě a veřejné správě*, Univerzita Pardubice, 2012
- [35] Teorie grafů. <http://teorie-grafu.cz>. [Online; navštíveno 16. 3. 2017]
- [36] Základní grafové algortimy. <http://algoritmy.eu/zga/pruchod-grafu/komponenty-2-souvislosti/>. [Online; navštíveno 15. 3. 2017]
- [37] Open Traveling Salesman Problem - Genetic Algorithm. <http://www.mathworks.com/matlabcentral/fileexchange/21196-open-traveling-salesman-problem-genetic-algorithm>. [Online; navštíveno 3. 3. 2017]
- [38] Fenclová, N.: *Diplomová práce: Nástroje pro řízení a vyhodnocování mikroindentačních zkoušek*, České vysoké učení technické v Praze, vedoucí práce Fíla, T., Kytýř, D., 2015

- [39] Krčmářová, N., Šleichrt, J., Doktor, T., Kytýř, D., Jiroušek, O.: *Semi-automated assessment of micromechanical properties of the metal foams on the cell-wall level*, 2017
- [40] LinuxCNC. <http://linuxcnc.org>. [Online; navštíveno 5. 5. 2017]
- [41] Encoder guide. <http://www.anaheimautomation.com/manuals/forms/encoder-guide.php#sthash.aKzZQKHg.5HqDlCls.dpbs>. [Online; navštíveno 5. 5. 2017]
- [42] Šleichrt, J.: *Diplomová práce: Modulární jednotka řízení experimentálních zařízení pro měření mechanických vlastností materiálů*, České vysoké učení technické v Praze, vedoucí práce Fíla, T., Kytýř, D., 2016
- [43] G-kód. <https://cs.wikipedia.org/wiki/G-kód>. [Online; navštíveno 28. 2. 2017]
- [44] LinuxCNC G code Tutorial. <http://gnipsel.com/linuxcnc/g-code/gen01.html>. [Online; navštíveno 24. 3. 2017]
- [45] G code Tutorial. <http://gnipsel.com/linuxcnc/g-code/index.html>. [Online; navštíveno 25. 3. 2017]
- [46] CNC G Code Programming: A CNC Mill Tutorial explaining G Codes. <https://www.youtube.com/watch?v=hq47wq9qBbs>. [Online; navštíveno 23. 3. 2017]
- [47] G-Code Lesson 1 What is G-Code? <https://www.youtube.com/watch?v=r5bimWKeMbY>. [Online; navštíveno 23. 3. 2017]

Příloha A - Matlab

Export

```
function [ ] = Export( results ,meritko ,referencni_prvni_obraz , start )

global num2
num2 = 1;
global resultsvzdalenosti

figure( 'MenuBar ', 'none ', 'Name ', 'Export_dat ', 'NumberTitle ', 'off ',
'Visible ', 'on ', 'Position ', [100 ,50 ,800 ,680]);

%% uzivatelske menu pro export
hlist2= uicontrol( 'Style ', 'listbox ', 'max ', 1, 'min ', 1,
'Position ', [580,80,100,100], 'Callback ', @list_CallBack ,
'string ', { 'Geneticky_algoritmus '; 'Zravy_algoritmus ' });
table1 = uitable( 'ColumnName ', { 'x', 'y' }, 'Position ', [0,300,200,300]);
hname = uicontrol( 'Style ', 'text ', 'String ', 'Souradnice_indentu ',
'Position ', [0,600,200,15]);
table2 = uitable( 'ColumnName ', { 'x', 'y' }, 'Position ', [250,300,200,300]);
hname2 = uicontrol( 'Style ', 'text ', 'String ',
'Souradnice_indentu_po_optimalizaci ',
'Position ', [250,600,200,15]);
set( table1 , 'Data', results )
hoptimalization = uicontrol( 'Style ', 'pushbutton ', 'String ',
'Execute ', 'Position ', [680,520,100,25], 'Callback ', @opt_Callback );
```

```

hsave = uicontrol('Style','pushbutton','String','Save','Position',
[680,480,100,25],'Callback',@save_Callback);
hexit = uicontrol('Style','pushbutton','String','Exit','Position',
[680,440,100,25],'Callback',@exit_Callback);

%% polozka v listu - slouzi pro vyber funkce pro optimalizaci poradi indentu
function list_Callback(varargin)
    num2 = get(hlist2, 'Value');
end

%% optimalizace
function opt_Callback(varargin)

% vyber dvou moznosti optimalizace
userconfig = struct('showProg',false,'showResult',false,'showWaitbar',
false,'xy',results);

switch num2
    case 1
        vysledky=tspo_ga(userconfig);
        dilcivysledky=vysledky.xy;
        results=zeros(size(dilcivysledky,1),size(dilcivysledky,2));
        for i=1:length(vysledky.optRoute)
            results(i,:)=dilcivysledky(vysledky.optRoute(i),:);
        end
    case 2
        results=hladovy(results);
        otherwise
            disp('neco_je_spatne')
        end

set(table2, 'Data', results)

```

```

results=results*meritko;%prevedeni pozic z px na mm
results(:,1)=(results(:,1)+start(1)-referencni_prvni_obraz(1)*meritko);
%zapocitani odrezu
results(:,2)=(results(:,2)+start(2)-referencni_prvni_obraz(2)*meritko);
%zapocitani odrezu
end

```

```

%% save - generovani g kodu

```

```

function save_Callback(varargin)

```

```

%%g kod

```

```

FilterSpec = { '*.txt' };

```

```

[FileName,PathName] = uinputfile(FilterSpec);

```

```

cd(PathName);

```

```

Title = 'Save_table';

```

```

sp=horzcat(PathName,FileName);

```

```

fileID = fopen(sp, 'w');

```

```

fprintf(fileID, '%s\r\n', 'O<automated_positioning>_sub');

```

```

fprintf(fileID, '%s\r\n', 'M125');

```

```

fprintf(fileID, '%s\r\n', 'M123');

```

```

%souradnice

```

```

for i=1:size(results,1)

```

```

    fprintf(fileID, '%s_%s%f%s_%s%f%s\r\n', 'G0',

```

```

        'x[',results(i,1),'+#5168]', 'y[',results(i,2),'-#5169]');

```

```

    fprintf(fileID, '%s\r\n', 'O<indentation_instrumented>_call');

```

```

    fprintf(fileID, '%s\r\n', 'G90');

```

```

end

```

```

fprintf(fileID, '%s\r\n', 'M124');

```

```

%snimky

```



```

for i=1:size(results,1)
    fprintf(fileID , '%s_%s%f_%s%f\r\n', 'G0', 'x', results(i,1),
        'y', results(i,2));
    fprintf(fileID , '%s\r\n', 'M126');
end

```

```

fprintf(fileID , '%s\r\n', 'M2');
fclose(fileID);
end

```

```

%% exit
function exit_Callback(varargin)
    close
end
end

```

findsubmat [33]

obsah této funkce lze nalézt v citovaném zdroji

gui

```

function gui
% Create the UI as it is being constructed.
clc
clear
state = 1;
global image
global results
global image2
global referencni_prvni_obraz
global start
global velikost_obrazu

```

```
figure('MenuBar','none','Name','Hledani_rozmisteni_indentu','NumberTitle',
'off','Visible','on','Position',[100,50,800,680]);
```

```
%vytvoreni objektu
```

```
hwindow = axes('Units','pixels','Position',[20,250,650,380]);%okno img
```

```
settings = uicontrol('Style','pushbutton','String','Calibration',
'Position',[680,640,100,25],'Callback',@settings_Callback);
```

```
hload_image = uicontrol('Style','pushbutton','String','Load_image',
'Position',[680,600,100,25],'Callback',@load_image_Callback);
```

```
%nacist obrazek
```

```
hsave = uicontrol('Style','pushbutton','String','Export',
'Position',[680,400,100,25],'Callback',@save_Callback);%ulozit
```

```
hexit = uicontrol('Style','pushbutton','String','Exit',
'Position',[680,360,100,25],'Callback',@exit_Callback);%end
```

```
hadd = uicontrol('Style','pushbutton','String','Add_point',
'Position',[680,480,100,25],'Callback',@add_Callback);%pridat teziste
```

```
hdelete = uicontrol('Style','pushbutton','String','Delete_point',
'Position',[680,440,100,25],'Callback',@delete_Callback);%odebrat teziste
```

```
hname = uicontrol('Style','text','String','Name','Position',
[20,650,650,15]);%nazev zobrazeneho obrazku
```

```
TH = uicontrol('Style','slider','Position',[20,120,200,25],
'value',0,'min',0,'max',1,'CallBack',@TH_Callback);
```

```
Open_strel = uicontrol('Style','slider','Position',
[300,120,200,25],'value',0,'min',0,'max',30,'CallBack',
@Open_strel_Callback);
```

```
Close_strel = uicontrol('Style','slider','Position',
[300,50,200,25],'value',0,'min',0,'max',30,'CallBack',
@Close_strel_Callback);
```

```
BW = uicontrol('Style','slider','Position',[20,50,200,25],
'value',0,'min',0,'max',30,'CallBack',@BW_Callback);
```

```
inverse = uicontrol('Style','slider','Position',[690,50,50,25],
'value',0,'min',0,'max',1,'CallBack',@in_Callback);
```

```
hnameTH = uicontrol('Style','text','String',
```

```

'Threshold_level', 'Position', [20, 150, 200, 25]);
hnameOS = uicontrol('Style', 'text', 'String',
'Morphological_opening_level', 'Position', [300, 150, 200, 25]);
hnameCS = uicontrol('Style', 'text', 'String',
'Morphological_closing_level', 'Position', [300, 80, 200, 25]);
hnameBW = uicontrol('Style', 'text', 'String',
'BW', 'Position', [20, 80, 200, 25]);
hnamein = uicontrol('Style', 'text', 'String',
'in', 'Position', [690, 80, 50, 25]);
hTHvalue = uicontrol('Style', 'text', 'String',
get(TH, 'Value'), 'Position', [230, 120, 25, 25]);
hOSvalue = uicontrol('Style', 'text', 'String',
get(Open_strel, 'Value'), 'Position', [510, 120, 25, 25]);
hCSvalue = uicontrol('Style', 'text', 'String',
get(Close_strel, 'Value'), 'Position', [510, 80, 25, 25]);
hBWvalue = uicontrol('Style', 'text', 'String',
get(BW, 'Value'), 'Position', [230, 80, 25, 25]);
hinvalue = uicontrol('Style', 'text', 'String',
get(inverse, 'Value'), 'Position', [750, 80, 25, 25]);
hshow= uicontrol('Style', 'pushbutton', 'String',
'Show_binary_image', 'Position', [680, 560, 100, 25], 'Callback', @show_Callback);
hexecute = uicontrol('Style', 'pushbutton', 'String',
'Execute', 'Position', [680, 520, 100, 25], 'Callback', @execute_Callback);
%provest skript
hlist= uicontrol('Style', 'listbox', 'max', 1, 'min', 1,
'Position', [580, 80, 100, 100], 'Callback', @list_Callback,
'string', {'Teziste'; 'Najdi_matici'; 'Nafukovani'});
hvel = uicontrol('Style', 'text', 'String', 'Velikost_pixelu [um]:
', 'Position', [20, 200, 350, 15]); %nazev zobrazeneho obrazku
hvel2 = uicontrol('Style', 'text', 'String', '-', 'Position',
[350, 200, 350, 15]); %nazev zobrazeneho obrazku

%% get values for thresholding

```

```

function TH_CallBack(varargin)
    num = get(TH, 'Value');
    set(hTHvalue, 'String', num2str(num));
end

```

```

function in_CallBack(varargin)
    num = ceil(get(inverse, 'Value'));
    set(hinvalue, 'String', num2str(num));
end

```

```

function Open_strel_CallBack(varargin)
    num = ceil(get(Open_strel, 'Value'));
    set(hOSvalue, 'String', num2str(num));
end

```

```

function Close_strel_CallBack(varargin)
    num = ceil(get(Close_strel, 'Value'));
    set(hCSvalue, 'String', num2str(num));
end

```

```

function BW_CallBack(varargin)
    num = ceil(get(BW, 'Value'));
    set(hBWvalue, 'String', num2str(num));
end

```

%% settings – slouzi k zmereni merky, zjistení skutečné velikosti 1 px

```

function settings_Callback(varargin)
    myload();
    prompt = {'Zadej_delku_merky:'};
    dlg_title = 'Realna_velikost';
    num_lines = 1;
    defaultans = {'0.400'};
    answer = inputdlg(prompt, dlg_title, num_lines, defaultans);
    velikostmerky=str2num(answer{1})

```

```

meritka=[0,0,0,0,0];
choice = questdlg
('Oznac_v_obraze_velikost_merky_stanovenou_v_predchozim_kroku_(5x)',
'Oznac','ok','ok');

for i=1:5
    [a,b] = ginput(2); %nacteme dva vstupy z mysí
    meritkodilci=abs(velikostmerky/(sqrt((b(2)-b(1))^2+(a(2)-a(1))^2)));
    %vypocet vzdalenosti
    meritka(i)=meritkodilci;
end

meritko=sum(meritka)/5;%vypocet prumerne vzdalenosti
hvel2 = uicontrol('Style','text','String',meritko,'Position',
[350,200,350,15]);
save pxsize.txt meritko -ascii

end

%% load image
function load_image_Callback(varargin)
[image,FileName, level, referencni_prvni_obraz, start, pocetradku,
pocetsloupcu, posun]=skladani_obrazu();
hname = uicontrol('Style','text','String',FileName,'Position',
[20,650,650,15], 'Callback',@name_Callback);
set(TH, 'value', level);
set(hTHvalue, 'String', level);

%%save

FilterSpec = {'*.png'};
[FileName,PathName] = uiputfile(FilterSpec);

```

```

sp=horzcat(PathName,FileName);
fileID = fopen(sp, 'w');
f=figure;
imshow(image);

hold on
for i=1:pocetradku
    line([0,pocetsloupce*posun/2],[i*posun/2 i*posun/2], 'Color','red')
        %vykresluje hranice dilcich obrazu
end

for i=1:pocetsloupce
    line([i*posun/2 i*posun/2],[0,pocetradku*posun/2], 'Color','red')
end

saveas(f, sp, 'png')
close(f)

fclose(fileID);
end

%% show binary image
function show_Callback(varargin)
THlevel=get(TH, 'Value');
openStrelSize=ceil(get(Open_strel, 'Value'));
closeStrelSize=ceil(get(Close_strel, 'Value'));
bwAreaSize=ceil(get(BW, 'Value'));
inverse_value=ceil(get(inverse, 'Value'));
image2=show_binary(image, THlevel, openStrelSize, closeStrelSize,
    bwAreaSize, inverse_value);
end

%% listbox

```

```

function list_CallBack(varargin)
    state = get(hlist, 'Value');
end

%% execute

function execute_Callback(varargin)
    THlevel=get(TH, 'Value');
    openStrelSize=ceil(get(Open_strel, 'Value'));
    closeStrelSize=ceil(get(Close_strel, 'Value'));
    bwAreaSize=ceil(get(BW, 'Value'));
    inverse=ceil(get(inverse, 'Value'));
    image2=show_binary(image, THlevel, openStrelSize,
    closeStrelSize, bwAreaSize, inverse);

    switch state
        case 1
            [stred1, stred2]=teziste(image2);
        case 2
            [stred1, stred2]=submat(image2);
        case 3
            [stred1, stred2]=nafukovani(image2);
        otherwise
            disp('neco_je_spatne');
            end
            results=[stred2 ' stred1'];
            velikost_obrazu=size(image2)

end

%% add pridani bodu

function add_Callback(varargin)
    [x,y] = ginput(1);

```

```

    plot(x,y,'g+', 'MarkerSize', 10);
    results=[results;x y];
end

%% delete - odebirani bodu
function delete_Callback(varargin)
[x,y] = ginput(1);
for i=1:size(results,1)
    if abs(results(i,1)-x)<15 && abs(results(i,2)-y)<15
        delete_line=i;
        results=[results(1:delete_line-1,:);
        results(delete_line+1:size(results,1),:)];
        break
    end
end

imshow(image2);
hold on

for i=1:size(results,1)
    plot(results(i,1),results(i,2),'g+', 'MarkerSize', 10);
end
end

%% exit
function exit_Callback(varargin)
    close
end

%% export
function save_Callback(varargin)
results(:,1)=(velikost_obrazu(2)-results(:,1));
results(:,2)=(velikost_obrazu(1)-results(:,2));

```



```
results=2*results; %prevedeni vysledku na puvodni hodnoty  
(obrazy jsou puvodne zredukovany na mensi velikosti)  
Export(results , meritko , referencni_prvni_obraz , start)
```

```
end
```

```
end
```

hladovy

```
function [ results ] = hladovy( results )
```

```
dilcivysledky=results;  
aktualnibod=dilcivysledky(1,:)  
results = [aktualnibod];  
pracovnimatice=dilcivysledky(2: size(dilcivysledky ,1) ,:);
```

```
while size(pracovnimatice ,1)>0
```

```
    for i=1:size(pracovnimatice ,1)
```

```
        maticevzdalenosti(i)=sqrt((pracovnimatice(i ,1)  
        -aktualnibod(1,1))^2+(pracovnimatice(i ,2)  
        -aktualnibod(1 ,2))^2);
```

```
    end
```

```
[minvalue , minindex]=min(maticevzdalenosti);  
results=[results ; pracovnimatice(minindex ,:)];  
aktualnibod=pracovnimatice(minindex ,:);  
pracovnimatice(minindex ,:)=[];  
maticevzdalenosti = [];
```

```
end
```

kalibrace

```
function [ pravyodrez , horniodrez ] = kalibrace( obraz1 , obraz2 , obraz3 )
```

```
pracovniobraz=[obraz2 obraz1];
```

```

    imshow(pracovniobraz);
    choice1 = questdlg('Vyber_posun_v_ose_x',
    'Posun_v_ose_x', 'ok', 'nechci', 'nechci');
    switch choice1
        case 'ok'

status=0
while status==0
    pracovniobraz=[obraz2 obraz1];
    imshow(pracovniobraz);
    [x,y]=ginput(2);
    rozdil=abs(x(2)-x(1));
    mpracovniobraz=[obraz2(:,(1:size(obraz2,2)-rozdil)), obraz1];
    imshow(mpracovniobraz);
    choice = questdlg('V_poradku?', 'V_poradku', 'ok', 'nok', 'nok');
    %ukazuje obraz po te co je cast odrizunuta
    switch choice
        case 'ok'
            status = 1;
        case 'nok'
            status = 0;
    end
    end
    case 'nechci'
        rozdil=0;
end

pracovniobraz=[obraz3; obraz1];
imshow(pracovniobraz);
choice2 = questdlg('Vyber_posun_v_ose_y',
'Posun_v_ose_y', 'ok', 'nechci', 'nechci');
switch choice2
    case 'ok'

```

```

        status=0
        while status==0
            pracovniobraz=[obraz3; obraz1];
            imshow( pracovniobraz );
            [x,y]=ginput(2);
            rozdil2=abs(y(2)-y(1));
            mpracovniobraz=[obraz3((1:size(obraz3,1)-rozdil2),:);
            obraz1];
            imshow(mpracovniobraz);

choice = questdlg( 'V_poradku?', 'V_poradku', 'ok', 'nok', 'nok' );
switch choice
    case 'ok'
        status = 1;
    case 'nok'
        status = 0;
    end
end
    case 'nechci'
        rozdil2=0;
    end
end

pravyodrez=rozdil
horniodrez=rozdil2

end

myload

function [ image, FileName, level ] = myload( varargin );

FilterSpec = { '*.tif' };
Title = 'Select_the_image';

```

```

[FileName,PathName] = uigetfile(FilterSpec,Title);
image=imread(horzcat(PathName,FileName));
    if size(image,3)==3
        image = rgb2gray(image);
    end
level = graythresh(image);
imshow(image);

end

```

nafukovani

```

function [ stred1 stred2 ] = nafukovani( image2 )

stred1=[]
stred2=[]
image=image2;
size(image)
for i=1:(size(image,1)-500)%cyklus v kazdem kole zvetsuje masku
    for j=1:(size(image,2)-500)
        for k=100:500
            maska=ones(k);
            vzor=image(i:i+k-1,j:j+k-1);

            if sum(maska)~=sum(vzor)
                k=0;
                break;
            end

            vzdalenost=0;
            podminka=0;
            for l=1:length(stred1)
                if abs(stred1(l)-i)<500 && abs(stred2(l)-j)<500

```

```

        vzdalenost=1;
        break;
    end
end

    if sum(vzor)==500*500
        podminka=1;
    end

        if podminka==1 && vzdalenost~=1
            stred1=[stred1 i+250];
            stred2=[stred2 j+250];
        end;
    end

end;

end;

imshow(image);
hold on
for i=1:length(stred1)
    plot(stred2(1,i),stred1(1,i),'r+', 'MarkerSize', 10);
end

end

show_binary

function [ image2 ] = show_binary( image, THlevel, openStrelSize,
                                closeStrelSize, bwAreaSize, inverse_value )

```

```

binarized = im2bw(image, THlevel);
    if inverse_value==1
        binarized=imcomplement(binarized);
    end;

step1 = binarized;
step2=imopen(step1, strel('disk', openStrelSize));
step3=imclose(step2, strel('disk', closeStrelSize));
image=step3;
image2=bwareaopen(image, ceil(bwAreaSize));
imshow(image2)

end

```

skladani_obrazu

```

function [ image, FileName, level, referencni_prvni_obraz, start,
pocetradku, pocetsloupcu, posun ] = skladani_obrazu( varargin )
clc
clear

FilterSpec = { '*.png' };
Title = 'Select_the_image_directory';
[FileName, PathName] = uigetfile(FilterSpec, Title);
%dostaneme jmeno a adresu souboru
cd(PathName);
jointdir = [dir('*.png')];
files=jointdir;

% Read all image files
for i=1:size(files, 1)%
%cyklus na prevadeni z rgb do grey
    image=imread(files(i).name);

```

```

        if size(image,3)==3
            image = rgb2gray(image);
        end
        original(:,:,i)=image;%nacteme i-ty obrazek
    end

    %nastaveni rozmeru vyslednyho obrazu
    prompt = {'Pocet_radku: ', 'Pocet_sloupcu: ',
        'Zadej_velikost_dilcich_obrazu ', 'koeficient ', 'x ', 'y '};
    dlg_title = 'Input';
    num_lines = 1;
    defaultans = {'14', '17', '800', '1', '17.104555', '8.746827'};%
    answer = inputdlg(prompt, dlg_title, num_lines, defaultans);

    posun=str2num(answer{3})
    koeficient=str2num(answer{4})
    posun=posun*koeficient;

    pocetradku=str2num(answer{1});
    pocetsloupcu=str2num(answer{2});
    start=[str2num(answer{5}),str2num(answer{6})];
    % urceni stredy obrazu
    [y,x]=size(original(:,:,1));
    stred=[x/2,y/2];

    result = [];
    obraz1=original(:,:,1);
    obraz1=obraz1(round(y/2-(posun-1)/2):round(y/2+(posun-1)/2),
    round(x/2-(posun-1)/2):round(x/2+(posun-1)/2));
    obraz2=original(:,:,2);
    obraz2=obraz2(round(y/2-(posun-1)/2):round(y/2+(posun-1)/2),
    round(x/2-(posun-1)/2):round(x/2+(posun-1)/2));
    obraz3=original(:,:, pocetsloupcu+1);

```

```

obraz3=obraz3(round(y/2-(posun-1)/2):round(y/2+(posun-1)/2),
round(x/2-(posun-1)/2):round(x/2+(posun-1)/2));
%vybere obrázky, které jsou pak použity pro kalibraci

[pravyodrez, horniodrez] = kalibrace( obraz1, obraz2, obraz3 );
referencni_prvni_obraz=[posun/2-pravyodrez, posun/2-horniodrez ]
%vytvareni velkyho obrazu, k jsou radky, j jsou sloupce
for k=1:pocetradku
    id=(k-1)*pocetsloupce+1;
    for j=id:id+pocetsloupce-1

        image=original(:, :, j); %vybere obraz
        vyrez=image(round(y/2-(posun-1)/2):
round((y/2+(posun-1)/2-horniodrez)),
round(x/2-(posun-1)/2):round((x/2+(posun-1)/2)-pravyodrez));

        %udela vyrez
        vyrez=imresize(vyrez,0.5);% priprava na redukci velikosti
        result=[vyrez result]; %vlozi vyrez do radku

    end
if k>1
    final=[result;final];
else
    final=result;
end
result = [];
end
rozmer=size(final)
image=final;
level = graythresh(image);

    imshow(image);

```


end

submat

```
function [ stred1 ,stred2 ] = submat( image2 )
```

```
image=image2 ;
```

```
B=ones (500);
```

```
[ a ,b]=findsubmat (image ,B)
```

```
%%stredy
```

```
stred1=[a (1)];
```

```
stred2=[b(1)];
```

```
for i=2:length(a)
```

```
    for k=1:length(stred1)
```

```
        vzdalenost=0;
```

```
        if abs(a(i)-stred1(k))<500 && abs(b(i)-stred2(k))<500
```

```
            %hleda blizke indenty
```

```
                break
```

```
        else vzdalenost=1;
```

```
        end
```

```
end
```

```
if vzdalenost==1
```

```
    stred1=[stred1 a(i)];
```

```
    stred2=[stred2 b(i)];
```

```
end
```

```
end
```

```

imshow(image);
    hold on
for i=1:length(stred1)
plot(stred2(1,i),stred1(1,i),'r+', 'MarkerSize', 10);
end

end

teziste

function [ ffstred1 , ffstred2 ] = teziste( image2 )

superstred1=[];
superstred2=[];
image=image2;

rbinarized=image;
[x,y]=size(rbinarized)

for i=1:fix(x/100-5) %50 nahrazeno 1000
    for j=1:fix(y/100-5)
        impart=(rbinarized((1+(100*(i-1)):(100*(i)+500)),
            (1+(100*(j-1)):(100*(j)+500))));
        %vybere cast obrazu kde hleda teziste
        size(impart);
        binaryImage = true(size(impart));
        pocetbilych=sum(sum(impart));

        if pocetbilych/(size(impart,1)*size(impart,2))>0.85
        measurements = regionprops(binaryImage , impart , 'WeightedCentroid');
%hleda teziste
        a = measurements(1).WeightedCentroid;

```

```

superstred1=[superstred1 (100*(i-1))+a(1,1)]
superstred2=[superstred2 (100*(j-1))+a(1,2)]

end
end
end

fstred1=[superstred1(1)];
fstred2=[superstred2(1)];

%% odstranovani indentu, ktere jsou prilis blizcko
for i=2:length(superstred2)
    for k=1:length(fstred1)
        vzdalenost=0;

        if (abs(superstred1(i)-fstred1(k))<600 &&
            abs(superstred2(i)-fstred2(k))<600)
            break
        else vzdalenost=1;
        end
    end
end

if vzdalenost==1
    fstred1=[fstred1 superstred1(i)];
    fstred2=[fstred2 superstred2(i)];
end
end

ffstred1=[];
ffstred2=[];

% odstraneni indentu, ktere jsou u okraje materialu

```

```

for i=1:length(fstred1)
    %%nejblizsi hrana
    [ii, jj] = find(rbinarized==0);
    d = sqrt((ii-fstred1(i)).^2+(jj-fstred2(i)).^2);
    blizko=min(d)
    %%konec hran
    if blizko > 100
        ffstred1=[ffstred1 fstred1(i)];
        ffstred2=[ffstred2 fstred2(i)];
    end
end
end

imshow(rbinarized);
hold on
for i=1:length(ffstred1)
    plot(ffstred2(1,i),ffstred1(1,i),'g+', 'MarkerSize', 10);
end
end

tspo_ga [37]

```

obsah této funkce lze nalézt v citovaném zdroji

Příloha B - G-kód

```
O<automated_positioning> sub
M125
M123
G0 x[12.453845+#5168] y[18.100904+#5169]
O<indentation_instrumented> call
G90
G0 x[12.012698+#5168] y[18.321478+#5169]
O<indentation_instrumented> call
G90
G0 x[11.599123+#5168] y[18.100904+#5169]
O<indentation_instrumented> call
G90
G0 x[11.323407+#5168] y[17.687329+#5169]
O<indentation_instrumented> call
G90
G0 x[11.406122+#5168] y[16.998038+#5169]
O<indentation_instrumented> call
G90
G0 x[12.132175+#5168] y[16.483367+#5169]
O<indentation_instrumented> call
G90
G0 x[12.582513+#5168] y[16.639606+#5169]
O<indentation_instrumented> call
G90
G0 x[13.290185+#5168] y[15.922743+#5169]
```

```
O<indentation_instrumented> call
G90
G0 x[13.060421+#5168] y[15.628645+#5169]
O<indentation_instrumented> call
G90
M124
G0 x12.453845 y18.100904
M126
G0 x12.012698 y18.321478
M126
G0 x11.599123 y18.100904
M126
G0 x11.323407 y17.687329
M126
G0 x11.406122 y16.998038
M126
G0 x12.132175 y16.483367
M126
G0 x12.582513 y16.639606
M126
G0 x13.290185 y15.922743
M126
G0 x13.060421 y15.628645
M126
M2
```