

Systém pro vektorizaci digitalizovaných grafů

Petr Přindiš

Bakalářská práce
2015



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2014/2015

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Petr Přindiš**
Osobní číslo: **A12054**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**
Forma studia: **prezenční**

Téma práce: **Systém pro vektorizaci digitalizovaných grafů**
Téma anglicky: **A System for the Vector Image Tracing of Scanned Charts**

Zásady pro vypracování:

1. Prostudujte systémy pro vektorizaci bitmapových křivek s otevřenou licencí.
2. Prozkoumejte vzorky digitalizovaných grafů (např. viskozita polymerů při tváření).
3. Analyzujte knihovny pro zpracování obrazů s otevřenou licencí a zvažte vhodnost jejich použití ve vaší práci.
4. Navrhněte algoritmy pro vektorizaci digitalizovaných grafů s jednou a více křivkami.
5. Navržené algoritmy implementujte a otestujte na dodaných vzorových grafech.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. HLAVÁČ, Václav a Milan ŠONKA. Počítačové vidění. Praha: Grada, 1992, 272 s. ISBN 80-854-2467-3.
2. DAWSON-HOWE, Kenneth. A practical introduction to computer vision with OpenCV. Wiley, 2014, pages cm. ISBN 978-111-8848-456.
3. BRADSKI, Gary R. Learning OpenCV. Sebastopol: O'Reilly, c2008, xvii, 555 s. ISBN 978-0-596-51613-0.
4. LAGANIÉRE, Robert. OpenCV 2 computer vision application programming cookbook: over 50 recepies to master this library of programming functions for real-time computer vision. 1st ed. Brimingham: Packt Publishing, 2011, iii, 287 s. Quick Answers to Common Problems. ISBN 978-1-84951-324-1.
5. CVonline: Visual Processing Software :Environments. FISHER, Robert. The University of Edinburgh [online]. 2014, 01/15/2015 [cit. 2015-01-27]. Dostupné z: <http://homepages.inf.ed.ac.uk/rbf/CVonline/SWEnvironments.htm>
6. PROSISE, Jeff. Programování ve Windows pomocí MFC. Vyd. 1. Praha: Computer Press, 2000, xxvi, 1135 s. ISBN 80-722-6309-9.

Vedoucí bakalářské práce:

Ing. Tomáš Dulík, Ph.D.

Ústav informatiky a umělé inteligence

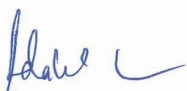
Datum zadání bakalářské práce:

6. března 2015

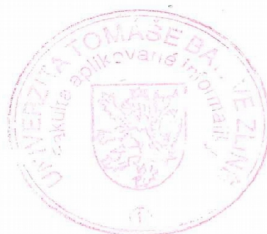
Termín odevzdání bakalářské práce:

22. května 2015

Ve Zlíně dne 6. března 2015



doc. Mgr. Milan Adámek, Ph.D.
děkan



L.S.



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl jsem seznámen s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnaní případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor;
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

ABSTRAKT

Hlavním cílem této bakalářské práce bylo navrhnout algoritmus pro sledování křivky v rastrovém obrázku za využití knihovny pro počítačové vidění s otevřenou licencí. Navrhnutý algoritmus byl následně použit k vytvoření aplikace pro digitalizaci naskenovaných, nebo jinak získaných obrázků grafů. Výsledná aplikace a sledovací algoritmus byly otestovány na dodaných vzorových grafech a jejich schopnosti porovnány s jinými programy umožňujícími digitalizaci křivek.

Klíčová slova: algoritmus, počítačové vidění, OpenCV, sledování čáry, digitalizace grafů

ABSTRACT

The main aim of this bachelor thesis was to design an algorithm for tracking curve in bitmap image using opensource library for computer vision. This algorithm was used to create application for digitizing scanned or otherwise acquired images of graphs. Application and tracking algorithm were tested on a set of sample graphs and their abilities compared with other softwares for digitizing curves.

Keywords: algorithm, computer vision, OpenCV, line tracking, graph digitizing

Chtěl bych poděkovat Ing. Tomáši Dulíkovi, Ph.D. za ochotu vést tuto práci. Dále bych chtěl poděkovat Dr. Jiřímu Švábíkovi za odbornou pomoc a za cenné rady a připomínky.

OBSAH

ABSTRAKT.....	5
ABSTRACT.....	5
ÚVOD.....	9
I TEORETICKÁ ČÁST.....	10
1 SOUČASNÉ ŘEŠENÍ PROBLÉMU.....	11
2 NÁSTROJE TŘETÍCH STRAN.....	11
2.1 PLOT DIGITIZER 2.0.....	11
2.2 CURVESNAP V1.0.....	13
2.3 GRAPH DIGITIZER SCOUT 1.2.4.....	14
2.4 DIGITIZEIT 2.0.....	15
3 POŽADAVKY.....	17
3.1 VZOROVÉ GRAFY.....	18
4 KNIHOVNY PRO POČÍTAČOVÉ VIDĚNÍ.....	18
4.1 VLFEAT.....	18
4.2 OPENCV.....	19
4.3 BOOFCV.....	20
4.4 AForge.NET.....	21
II PRAKTICKÁ ČÁST.....	22
5 REALIZACE VLASTNÍHO NÁSTROJE.....	23
5.1 TECHNICKÉ POŽADAVKY.....	23
5.2 VOLBA KNIHOVNY A VÝVOJOVÝCH NÁSTROJŮ.....	23
5.3 VÝVOJ PRVNÍ ČÁSTI APLIKACE.....	23
5.3.1 Struktura aplikace.....	23
5.3.2 Vykreslování pomocí GDI+.....	24
5.3.3 Použití dll knihovny.....	25
5.4 VÝVOJ DRUHÉ ČÁSTI APLIKACE.....	25
5.4.1 Úprava obrázku a filtrace.....	26
5.4.2 Nalezení oblasti křivky.....	27
5.4.3 Zmapování nalezené oblasti.....	27
5.4.4 Nalezení křivky.....	31
5.4.5 B-Spline křivky.....	32
5.5 OVLÁDÁNÍ APLIKACE.....	33
5.6 POROVNÁNÍ S KONKURENČNÍMI NÁSTROJI.....	39
5.6.1 Testování programu Curvesnap V1.0.....	40
5.6.2 Testování programu Graph Digitizer Scout 1.2.4.....	40
5.6.3 Testování programu Digitize It 2.0.....	40

5.6.4 Testování vlastního programu Compuplast Digitizer.....	40
ZÁVĚR.....	42
SEZNAM POUŽITÉ LITERATURY.....	43
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	45
SEZNAM OBRÁZKŮ.....	46
SEZNAM TABULEK.....	48
SEZNAM PŘÍLOH.....	48

ÚVOD

Počítačové vidění patří mezi mladá odvětví výpočetní techniky. Snahou je umožnit počítačům zpracovávat obrazová data a získávat z nich informace. Pro člověka, jakožto tvora co se rodí s očima a využívá je jako svůj primární smysl je jednoduché orientovat se v obrázku. Rozpoznávání objektů a informací mu nečiní žádný problém, jelikož je to činnost, kterou provádí každodenně po celý svůj život. Nemusí být proto na první pohled zřejmé, jak složité je zpracování i jednoduchého obrázku pomocí počítačového programu. Obrázek je v počítači nejčastěji reprezentován dvourozměrným polem bodů, ve kterém má každý bod určenou číselnou hodnotu barvy. Nalezení užitečných informací vyžaduje složité postupy a algoritmy, které si musí umět poradit s různorodostí obrazových dat.

Cílem této práce je vytvořit program, který usnadní zpracování skenovaných, nebo jinak získaných bitmapových obrázků grafů. Program umožní převést křivky z obrázku do číselné podoby, která je nutná pro další počítačové zpracování. Vzhledem k předpokladu častého používání je snaha o co možná nejjednodušší ovládání a minimalizování stráveného času. To zahrnuje hlavně automatické nalezení trasy zvolené křivky v obrázku.

V teoretické části práce se věnuji již existujícím nástrojům, které umožňují digitalizování bitmapových křivek, jejich možnostem a způsobu použití. Dále se věnuji některým vybraným knihovnám z oblasti zpracování obrazu a počítačového vidění, které mohou usnadnit tvorbu programu pro digitalizaci křivek.

V praktické části se věnuji návrhu algoritmu, který dokáže najít cestu křivky v obrázku bez ohledu na její tvar, barvu, tloušťku, nebo počtu křížení s jinými čarami a který si poradí i s poškozením obrázku vznikajícím například špatným naskenováním. Tento algoritmus následně implementuji v samostatné aplikaci.

V závěru práce porovnávám kvalitu navrženého algoritmu se schopnostmi jiných programů určených pro digitalizování křivek. Jako zkušební data slouží dodané vzorové grafy, které je možné najít v příloze této práce.

I. TEORETICKÁ ČÁST

1 SOUČASNÉ ŘEŠENÍ PROBLÉMU

Program Virtual Extrusion Laboratory (dále pouze VEL) od firmy Compuplast International a.s. je světovou špičkou mezi softwarem pro simulaci a optimalizaci nejrůznějších procesů spojených s vytlačováním polymerů. Jelikož polymery prochází neustálým vývojem, jejich vlastnosti a chování jsou rozdílné. Pro správný průběh simulace je tedy mimo jiné potřeba znát vlastnosti požadovaného materiálu. Problémem nastává v okamžiku, kdy výrobce polymeru neposkytuje naměřená data v počítačem čitelné podobě, ale pouze ve formě grafu, například ve svém katalogu. Pro získání hodnot, které se mohou dále zpracovat je pak potřeba takovýto graf naskenovat, nebo vyfotit a poté použitím nástroje pro digitalizování křivek převést do podoby, kterou dokáže program VEL zpracovat. V současnosti firma Compuplast nemá vlastní nástroj, který by uživateli tento krok zjednodušil. Jedinou možností je využití některého z programů třetích stran, graf jím zpracovat a hodnoty ručně vložit do programu VEL. Tento způsob je značně neefektivní a proto vznikl požadavek na vytvoření softwaru, který by umožnil digitalizovat hodnoty přímo z prostředí programu VEL a značně tak zjednodušil práci. [1]

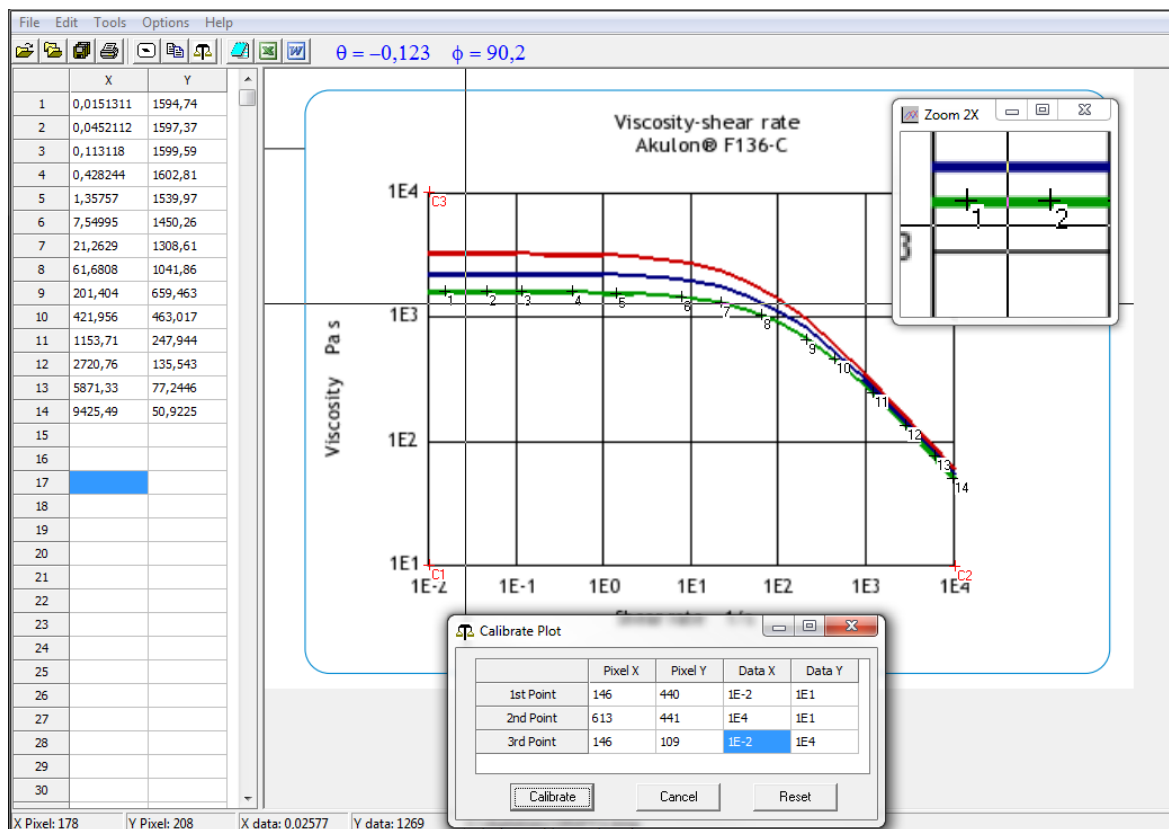
2 NÁSTROJE TŘETÍCH STRAN

Existuje spousta nástrojů pro digitalizaci bitmapových grafů. Velká část z nich je šířena zdarma jako freeware program volně dostupný na síti internet. Kvalita těchto programů je různá. V této práci porovnám několik z nich s ohledem na použitelnost pro digitalizování viskozitních křivek na vzorových grafech poskytnutých firmou Compuplast. Při výběru programů pro digitalizování křivek jsem vybíral z řad freeware i placených programů.

Všechny zde uvedené programy jsem zkoušel pod operačním systémem Windows 7 64 bit. V jiných verzích operačního systému Windows se mohou chovat odlišně.

2.1 Plot digitizer 2.0

Tento program je jednoduchý digitalizační nástroj vyvinutý pod katedrou Fyziky na univerzitě v Jižní Karolíně. Je možné jej zdarma stáhnout ze stránek univerzity [4].



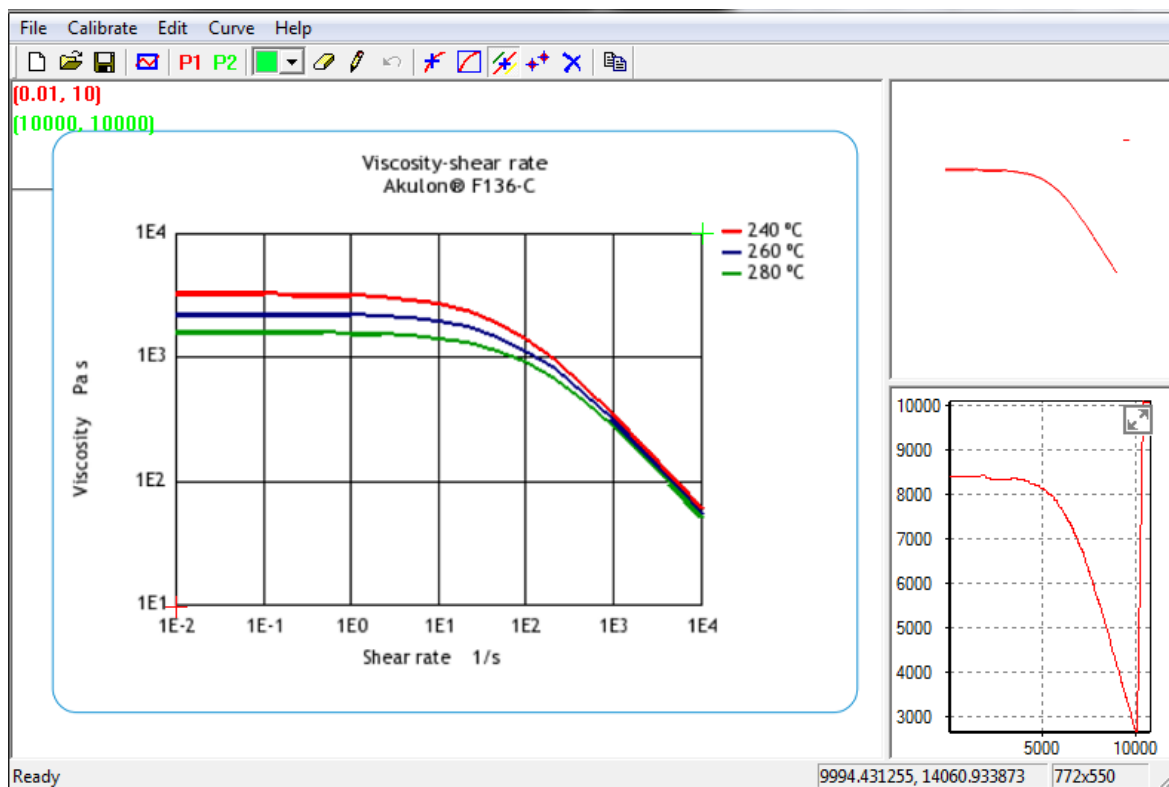
Obr. 1: Plot digitizer 2.0

Po spuštění programu se volbou *File* → *Open Image File* vybere obrázek pro digitalizování. Obrázek grafu je statický, nelze s ním manipulovat. V případě, že by byl obrázek otočený, značně by to znepříjemnilo práci. Jako druhý krok je potřeba nastavit typ grafu. Pod volbou *Options* → *Plot Type* je možné zvolit mezi lineárními a logaritmickými souřadnicemi. Dalším krokem je zvolení os X, Y a jejich rozsahů. K tomu slouží tabulka pod volbou *Tools* → *Calibrate Plot*. V této tabulce se nastavují tři body a jejich hodnoty, podle kterých se následně vypočítá souřadnicový systém. Vlastní digitalizování křivky se provádí postupným klikáním na křivku. Žádný automatický nástroj pro sledování křivky zde není. Hodnoty ručně zvolených bodů se vypisují v levé části aplikace, odkud je možné je přímo zkopírovat. Pro přesnější výběr bodů je zde okno, ve kterém se zobrazuje zvětšená oblast kolem kurzoru. Vybrané body není možné nijak přesouvat. Smazat bod je možné stisknutím pravého tlačítka myši v blízkosti bodu.

Tento program není příliš vhodný na digitalizování grafů s více křivkami, jelikož mezi body jednotlivých křivek nijak nerozlišuje. Nastavení souřadnicového systému je zbytečně složité a zdouhavé. Nevýhodou je i absence automatického stopování křivky.

2.2 CurveSnap V1.0

CurveSnap je zdarma dostupný nástroj pro digitalizování skenovaných grafů. Program je založený na knihovně pro počítačové vidění OpenCV 2.4.3, která mu umožňuje i jednoduché automatické sledování křivky grafu [5].



Obr. 2: CurveSnap V1.0

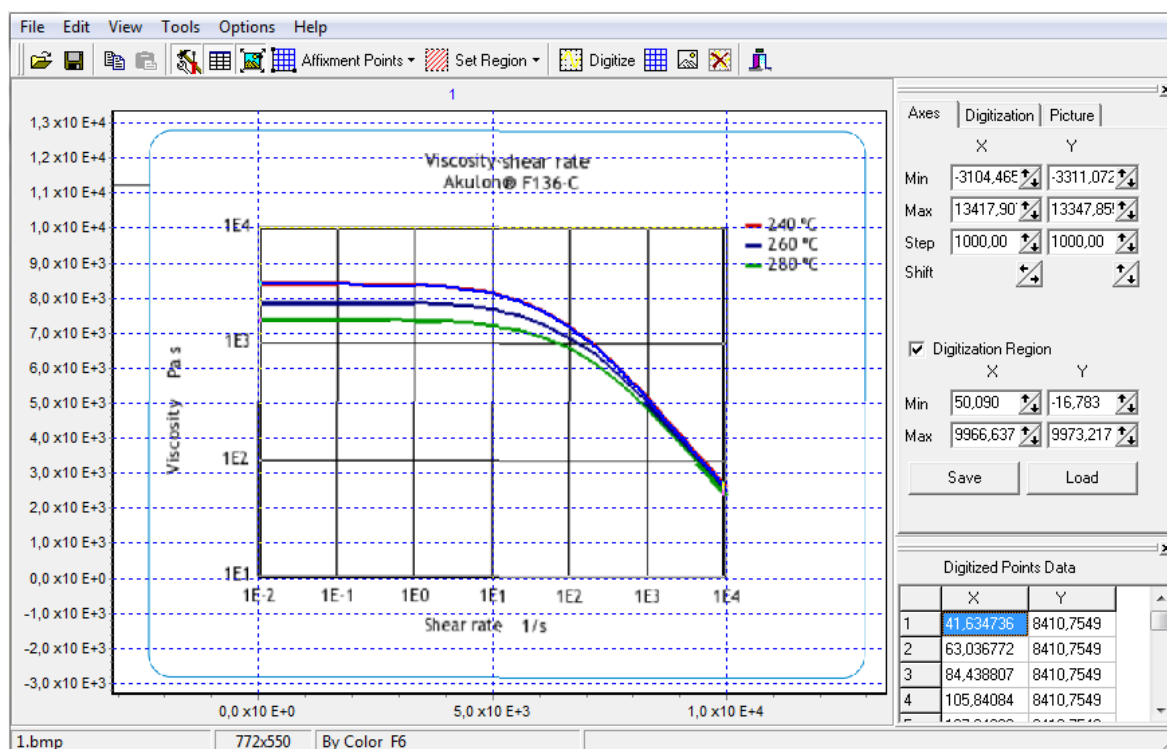
Obrázek grafu je možné nahrát dvěma způsoby. První je klasické otevření předem připraveného obrázku volbou *File* → *Open*. Druhou možností je získání grafu přímo z obrazovky počítače použitím volby *File* → *Snap*. Na obrazovce se poté tahem myši vyznačí obdélníková oblast obsahující graf a ten se následně zobrazí v okně programu. Souřadnicový systém se nastavuje pomocí dvou známých bodů, které se umístí do obrázku grafu volbou *Calibrate* → *Set Point*. Dále je možné upravit obrázek pomocí dvou nástrojů. Guma a tužka. Tímto se dají odstranit nepřesnosti v obrázku, které mohou způsobit špatnou detekci. Samotné zaznamenání křivky je možné provést čtyřmi způsoby nalézajícími se pod záložkou *Curve*. Nejjednodušší způsob *Choose points* umožňuje ručně zaznamenat jednotlivé body křivky. Pro automatickou detekci je potřeba zvolit pokročilejší nástroje. *Choose by connectivity* vybere spojitou oblast. Detekování křivky ve vzorovém grafu se tímto nástrojem nepodařilo. Další možností je detekování křivky podle barvy (*Choose by color*), které dopadlo na vzorovém grafu podstatně lépe a získaná data by již byla použitelná. Poslední nástroj *Choose Rectangle* najde křivku v obdélníku, který se vyznačí tahem myši.

Použití tohoto nástroje dalo velmi špatné výsledky, jelikož není možné označit obdélníkem jenom jednu křivku.

CurveSnap je zajímavý nástroj pro rychlé digitalizování jednoduchých grafů. Velkou výhodou je automatické sledování čáry, které zde demonstruje možnosti použité knihovny OpenCV. Nevýhodou je absence nastavení souřadnicového systému v logaritmických souřadnicích, které značně zkomplikuje digitalizování takovýchto grafů. Další nevýhodou je absence transformování obrázku, tj. jeho otáčení a přibližování. To může způsobit velké nepřesnosti při ručním zadávání bodů. Programu také chybí jakákoliv možnost editace nalezených, nebo ručně zvolených bodů.

2.3 Graph Digitizer Scout 1.2.4

Graph Digitizer Scout od výrobce softwarových nástrojů Bytescout je placený nástroj pro profesionální použití. Výrobce uvádí vhodnost použití toho programu pro digitalizování skenovaných grafů za účelem využití získaných dat pro různé simulace. Cena plné verze programu se pohybuje od 35\$ za verzi pro osobní použití do 59\$ za firemní licenci. Za účelem vyzkoušení programu výrobce nabízí 30 denní zkušební verzi zdarma. [23]



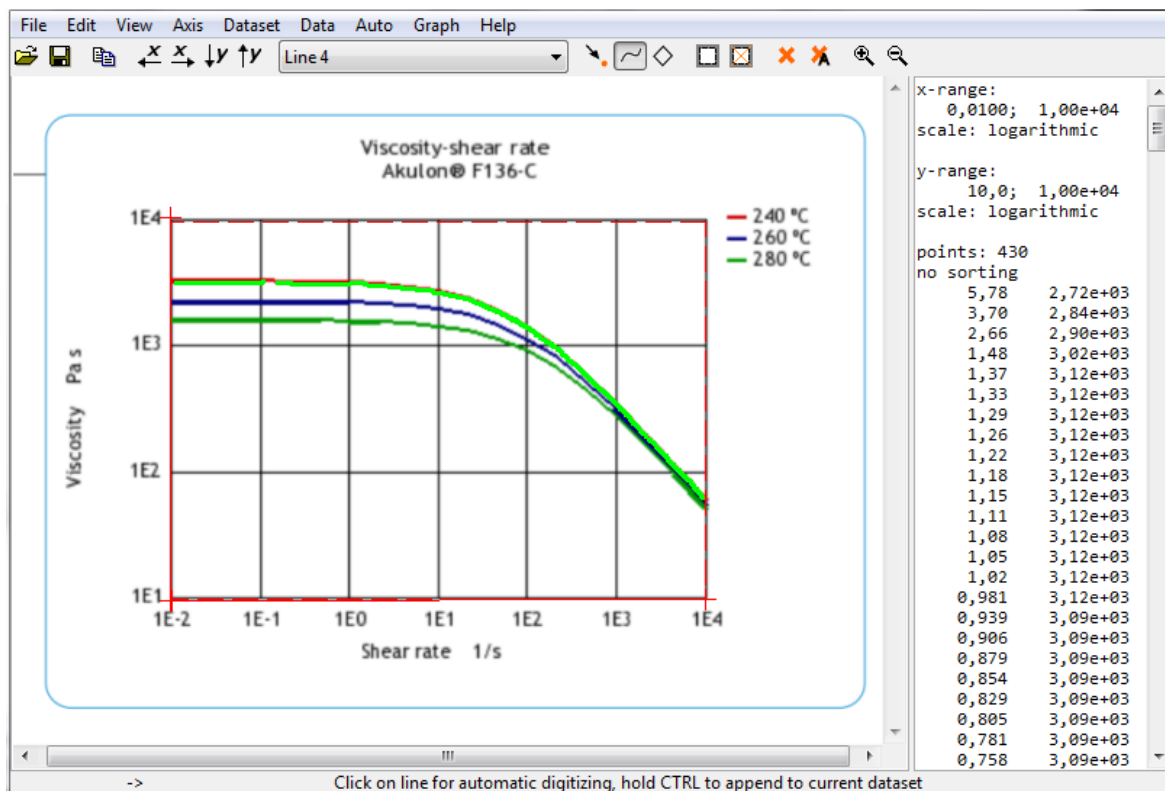
Obr. 3: Graph Digitizer Scout 1.2.4

Program se ovládá podobně jako předchozí programy. Po otevření obrázku s grafem se nastaví souřadnicový systém volbou *Tools* → *Bind Coordinates*. Nastavení se provede zadáním 2, 3, nebo 4 známých bodů. Následně se vybere oblast obrázku obsahující křivku volbou *Set Region*. Samotný obrázek je možné pro lepší čitelnost upravit změnou jasu, kontrastu, barevné sytosti a rozmazáním. Program také umožňuje obrázkem otáčet. Pro digitalizaci křivky jsou zde dostupné 3 režimy. Klasický ruční režim, ve kterém se body na křivce vyznačí ručně a 2 automatické. V automatickém režimu se zvolí digitalizování křivky buď podle barvy, nebo podle maximální světlosti. V režimu digitalizování podle barvy je potřeba nastavit barvu křivky dvojklikem na zvolenou křivku. Následně se tlačítkem *Digitize* provede vyhledání trasy křivky. Ve druhém režimu digitalizování je křivka hledána podle její světlosti. Předpokládá se, že křivka má jinou barvu, než pozadí, které je většinou bílé. Před digitalizováním je ale potřeba zaškrtnout volbu *Negative*, která invertuje barvy. Tím je zaručeno, že pozadí grafu bude tmavší, než požadovaná křivka. Po nalezení křivky některým z automatických režimů je možné volbou *Correcting* automaticky upravit šířku křivky. Digitalizované hodnoty se vypisují v pravé části okna aplikace, odkud se dají zkopírovat, nebo uložit do csv souboru.

Přesto že se jedná o placený program, je zde několik závažných nedostatků. Nejzávažnějším z nich je nemožnost nastavení logaritmických souřadnic. Dalším problémem je absence manipulace s obrázkem. Kromě rotace a roztáhnutí na celou obrazovku nejde s obrázkem nijak pohybovat. Program také nepodporuje digitalizování více křivek najednou. Je potřeba digitalizovat postupně a ukládat každou křivku zvlášť.

2.4 DigitizeIt 2.0

Další z řady placených nástrojů je program DigitizeIt. Licence pro neomezené použití stojí 49\$. Pro vyzkoušení programu výrobce nabízí 21 denní zkušební verzi, která je téměř shodná s plnou verzí. [8]



Obr. 4: DigitizeIt

Program DigitizeIt je už na první pohled propracovanější než předchozí porovnávané programy. Po otevření obrázku je možné s ním pohybovat, rotovat a zoomovat. Souřadnicový systém se nastaví tak, jako v předchozích programech 4 známými body. Pod volbou *Axis* je možné nastavit lineární, logaritmickou, nebo 1/x škálu. Program umožňuje, kromě ručního výběru bodů i automatické hledání křivky a to dvěma způsoby. První způsob slouží pro hledání spojitých křivek a po aktivaci stačí dvakrát poklepat na zvolenou křivku. V případě, že ji algoritmus nenajde celou, stačí podržením klávesy CTRL a dvojklikem na nevyznačenou část křivky doplnit zbylé body. Druhý způsob automatického hledání vyhledává podle zvoleného symbolu. Je tedy určen především na vyhledávání nespojitých křivek. Dvojklikem na zvolený symbol v grafu a nastavením přesnosti hledání program nalezne ostatní podobné body. Pokud je v grafu více křivek, stačí je postupně označovat a program si je sám roztřídí. To ale funguje pouze v automatickém režimu. Po nalezení všech žádaných křivek je před exportem dat možné seřadit body vzestupně, nebo sestupně.

K tomuto programu mám jen velmi málo výhrad. Obsahuje téměř všechny funkce, které bych od takového nástroje očekával. Jediným problémem může být složité zadávání os, které je zbytečně zdlouhavé. V některých případech to ale může být potřeba.

3 POŽADAVKY

Digitalizace grafů některým z nástrojů třetích stran je sice možná, ale kvalita a pohodlí při používání není příliš vysoká a to i u placených programů. Možnost používat vlastní program má několik nesporných výhod. Může být spouštěn přímo z programového prostředí VEL, se kterým přímo komunikuje, takže není potřeba data ručně kopírovat. Je možné k němu poskytnout klientům podporu a bezesporu největší výhodou je jednoduchá rozšiřitelnost o speciální funkce, kterými běžné nástroje tohoto druhu nedisponují.

Po konzultaci s pracovníky společnosti Compuplast vznikl seznam požadavků a vlastností, které by měl finální program obsahovat. Program bude vytvořen ve dvou verzích. První verze bude samostatná aplikace určená pro digitalizování obecných grafů. Druhá verze programu bude součástí prostředí VEL a bude speciálně určena pro digitalizování viskozitních grafů.

Vývoj programu je rozdělen do tří částí. V každé části jsou vytyčeny body, které by měl program splňovat.

1) První část

- a) Transformace bitmapového obrázku (posun, rotace, zoom).
- b) Nastavení souřadnicového systému.
- c) Ruční zaznačení bodů křivky.
- d) Podpora více křivek.
- e) Exportování dat v logaritmických souřadnicích.

2) Druhá část

- a) Automatická detekce křivky.
- b) Editace bodů křivky (posun, mazání).
- c) Proložení bodů B-spline křivkou.

3) Třetí část

- a) Prokládání bodů speciálními křivkami.
- b) Zobrazení fitované křivky pro vizuální porovnání s původními hodnotami.
- c) Další, dosud nespecifikované požadavky.

Tato práce se zabývá pouze prvními dvěma vývojovými body. Požadavky pro třetí část vývoje programu nejsou kompletní a budou teprve upřesněny podle potřeb, které vyplynou z používání prvních dvou verzí aplikace.

3.1 Vzorové grafy

Firma Compuplast International mi poskytla několik vzorových grafů pro testování digitalizéru v průběhu jeho vývoje. Většina grafů pochází z projektů, které tato firma zpracovávala a u kterých byl k dispozici pouze materiál, jehož vlastnosti byly popsány pouze křivkou v obrázku grafu. Jde tedy o reálná data, se kterými by si měl mnou vyvinutý program poradit. Záměrně byly zvoleny takové grafy, u kterých se předpokládají problémy při digitalizování. Všechny použité grafy jsou v příloze 1 k této práci.

Z dodaných obrázků grafů je patrné, jak velké mohou být rozdíly v grafickém zpracování dat od různých dodavatelů. Ve všech vzorových grafech se vyskytují různé prvky, jako jsou svislé a vodorovné vodící čáry, mřížky, nápisy a značky různých tvarů a velikostí. Některé grafy mají jednotlivé křivky barevně odlišené, u jiných jsou pouze jednobarevné. Často se v grafech požadovaná křivka kříží, nebo částečně překrývá s jinými čarami a křivkami. Výsledný algoritmus bude muset s těmito nedokonalostmi v grafech počítat a účinně je odfiltrovat.

4 KNIHOVNY PRO POČÍTAČOVÉ VIDĚNÍ

4.1 VLFeat

VLFeat je multiplatformní open source knihovna počítačového vidění. Hlavním zaměřením knihovny je detekce a vyhledávání objektů v obraze a vzájemné porovnávání snímků. Knihovna obsahuje aplikační rozhraní v jazyce C pro vývoj aplikací pod různými programovacími jazyky a rozhraní pro program Matlab. Funkce knihovny jsou rozděleny do několika částí, viz tabulka 1. [7]

Tab. 1. Popis částí knihovny VLFeat [7]

<i>Modul</i>	<i>Popis</i>
Visual feature detectors and descriptors	V této části se nachází algoritmy pro hledání a popis významných bodů v obrázku. Mezi tyto algoritmy patří např. MSER (Maximally Stable Extremal Regions), SIFT(Scale Invariant Feature Transform) a další.
Clustering and indexing	Zde jsou algoritmy jako např. K-Means sloužící ke shlučování objektů na základě jejich vlastností.
Segmentation	Algoritmy pro rozdělení obrázku na jednotlivé segmenty.
Statistical methods	Statistické algoritmy např. Homogeneous kernel map, nebo SVM (Support vector machines)
Utilities	Zde jsou různé algoritmy pro práci s obrázkem, matematické operace, generátor náhodných čísel a další.



Obr. 5: Spojení dvou snímků na základě výpočtu klíčových bodů algoritmem SIFT [7]

Projekt knihovny VLFeat byl založen v roce 2007 a je stále aktivně vyvíjen. Nejnovější verze v době psaní této práce byla 0.9.20. což potvrzuje, že práce na této knihovně ještě nejsou u konce. Velkou předností knihovny VLFeat je kvalitně zpracovaná dokumentace včetně spousty příkladů a ukázek kódu. [7]

4.2 OpenCV

OpenCV je opensource multiplatformní knihovna pro počítačové vidění šířená pod licencí BSD. Knihovna je napsaná v jazyce C a C++ což umožňuje fungování na všech v současné době nejpopulárnějších platformách. Podporované operační systémy jsou Windows, Linux, MacOS, FreeBSD, OpenBSD a mobilní systémy Android, Maemo, iOS. OpenCV dokáže

využít akceleraci výpočtů pomocí grafické karty od firmy NVIDIA za použití architektury CUDA. Díky tomu je možné dosáhnout výrazného zrychlení některých paralelních výpočtů. Další možností urychlení výpočtů je využití OpenCL akcelerace, kterou knihovna rovněž nabízí. Vývoj aplikací založených na této knihovně je možný kromě klasických jazyků C a C++ také v jazyce Python, Ruby, Matlab a dalších. [2, 3]

Knihovna OpenCV má modulární strukturu. Jednotlivé funkce OpenCV jsou logicky uspořádány do několika modulů. Popis dostupných modulů je v následující tabulce.

Tab. 2. Popis modulů OpenCV

<i>Modul</i>	<i>Popis</i>
core	Základní modul obsahující datové struktury a některé základní funkce.
imgproc	Modul pro základní zpracování obrazu. Umožňuje filtrovat a transformovat obraz, převádět mezi barevnými modely a další.
video	Modul, který umožňuje sledování pohybujících se objektů, dokáže rozpoznat popředí a pozadí, odhadnout pohyb.
calib3d	Obsahuje základní 3D algoritmy, umožňuje nastavení jedné, nebo více kamer.
features2d	Algoritmy pro detekci v obraze.
objdetect	Slouží pro hledání předem definovaných objektů v obraze, například tváře.
highgui	Slouží pro práci s obrazovými a video soubory.
gpu	Obsahuje knihovny využívající akcelerace grafickou kartou.

Knihovna OpenCV je jednou z nejpoužívanějších knihoven v oblasti počítačového vidění. Komunita okolo této knihovny čítá přes 48 tisíc vývojářů a počet stažení je více než 7 milionů. OpenCV kompletně pokrývá oblast počítačového vidění. Obsahuje více než 2500 optimalizovaných algoritmů. Jako podporu vývojářům nabízí tvůrci knihovny velmi podrobnou on-line dokumentaci, spoustu tutoriálů a diskusní fórum zaměřené na řešení problémů při vývoji aplikací. [9]

4.3 BoofCV

BoofCV je opensource knihovna pro počítačové vidění s výpočty v reálném čase. Knihovna je určena pro vývoj aplikací v programovacím jazyce Java. Hlavní využití této knihovny je v oblasti zpracování obrazu a detekování objektů, které může být využito díky rychlosti algoritmů například v robotice. BoofCV knihovna je šířena pod Apache licenci a ji možné

použit i pro komerční aplikace. Zdrojové kódy a zkompileované soubory knihovny jsou volně k dispozici na serverech Github.com a Sourceforge.net, kde jsou také pravidelně aktualizovány. [10, 11, 12].

4.4 Aforge.NET

Aforge.NET je framework určený pro počítačové zpracování obrazu a umělou inteligenci. Jak už z názvu vyplývá, hlavním programovacím jazykem pro tento framework je jazyk C#. Použití knihovny je vymezeno licencemi LGPL v3 a GPL v3. Pro podporu vývojářům je k frameworku dostupná rozsáhlá online i offline dokumentace. [18]

Framework Aforge.NET se skládá z několika knihoven, přičemž každá má specifickou funkci.

Tab. 3. Popis knihoven frameworku Aforge.NET [18]

<i>Knihovna</i>	<i>Funkce</i>
AForge.Imaging	Zpracování obrazu.
AForge.Vision	Funkce pro počítačové vidění.
AForge.Video	Zpracování videa.
AForge.Neuro	Výpočty pomocí neuronových sítí.
AForge.Genetic	Evoluční algoritmy.
AForge.Fuzzy	Výpočty využívající Fuzzy logiku.
AForge.Robotics	Knihovny s podporou pro robotické stavebnice.
AForge.Machine-Learning	Strojové učení.

Jak je z tabulky 3 patrné, framework Aforge.NET je opravdu všestranně využitelný. Je na něm založena spousta projektů, které sahají od detekování pohybujících se objektů ve videu až po neuronové sítě. [19]

II. PRAKTICKÁ ČÁST

5 REALIZACE VLASTNÍHO NÁSTROJE

5.1 Technické požadavky

Program pro digitalizování bitmapových křivek je požadován ve dvou identických verzích. První verzi je samostatně nespustitelná dynamicky linkovaná knihovna (dll) obsahující všechny potřebné algoritmy a dialogová okna. Tato dll knihovna bude spouštěna a ovládána z programu VEL. Druhou verzí je jednoduchá aplikace, která umožní spuštění digitalizačního nástroje z dll knihovny mimo prostředí programu VEL a umožní načítání a ukládání dat do uživatelem zvoleného umístění.

Vzhledem k nutnosti kompatibility kódu s programem VEL musí být digitalizační program napsán v jazyce C++. Uživatelské rozhraní musí být vytvořeno pomocí Microsoft Foundation Classes knihoven.

5.2 Volba knihovny a vývojových nástrojů

Omezení daná technickými požadavky programu značně redukuje počet použitelných knihoven pro zpracování obrazu. Knihovna musí podporovat programovací jazyk C++. Dalším omezením je licence, pod kterou je knihovna šířena. Vzhledem k použití knihovny v komerční aplikaci je akceptovatelná pouze licence, umožňující její bezplatné použití bez nutnosti zveřejnění zdrojového kódu celé výsledné aplikace. S ohledem na tyto potřeby jsem zvolil knihovnu OpenCV ve verzi 2.4.3. Tato knihovna vyhovuje všem požadavkům. Je jednoduše použitelná a má velmi rozsáhlou uživatelskou podporu.

Prostředí pro vývoj jsem zvolil Microsoft Visual Studio 2012, které umožňuje programování MFC aplikací.

5.3 Vývoj první části aplikace

Cílem první fáze vývoje aplikace pro digitalizování bitmapových křivek je vytvoření základního programu, který umožní ruční vyznačení bodů na křivce. Do tohoto základního programu bude poté v druhé části přidána možnost automatické detekce zvolené křivky.

5.3.1 Struktura aplikace

Jádro aplikace tvoří DLL knihovna Rdll_GraphDig.dll, která obsahuje několik tříd. Jejich popis je v tabulce 4.

Tab. 4. Popis tříd DLL knihovny

<i>Třída</i>	<i>Popis</i>
RGDWindow	Hlavní třída obsahující okno s obrázkem grafu. Obsahuje kompletní logiku digitalizéru.
MenuDialog1	Třída obsahující okno s ovládacími prvky aplikace.
CRGrDigDlg	Tato třída vytváří hlavní dialogové okno aplikace, ve kterém zobrazuje ono s obrázkem grafu a okno s ovládacími prvky. Jako jediná je přístupná vně knihovny a umožňuje výměnu dat mezi aplikací a dll knihovnou.

5.3.2 Vykreslování pomocí GDI+

Hlavní třída programu RGDWindow se mimo jiné stará o vykreslování obrázku grafu a dalších grafických prvků, jako jsou například osy a vyznačené body. O vykreslování se stará rozhraní GDI+. Toto rozhraní je standardní součástí operačního systému Windows a umožňuje vykreslování čar, různých tvarů, obrázků, textů a dalších objektů na výstupní zařízení, v tomto případě obrazovku počítače. [13]

Aby používání digitalizačního nástroje bylo pro uživatele pohodlné, musí mít možnost úpravy polohy obrázku. Ne vždy se totiž podaří získat obrázek grafu oříznutý a natočený správným směrem. Většina vyzkoušených nástrojů pro digitalizování grafů neumožňuje příliš dobře manipulovat s obrázkem, což práci znesnadňuje. Rozhraní GDI+ umožňuje jednoduše implementovat všechny druhy transformací pomocí transformační matice. Použitím těchto funkcí je umožněno uživateli obrázek grafu otáčet kolem jeho středu, posouvat tahem myši do všech stran a přibližovat, nebo oddalovat kolečkem na myši. Výhodou transformační matice je jednoduchý převod souřadnic v transformovaném obrázku do souřadnic v původním obrázku. Stačí k tomu využít vlastnost transformační matice a změněné souřadnice vynásobit maticí inverzní. Díky tomu je jedno, jak uživatel obrázek transformuje, jelikož se vždy získají hodnoty odpovídající původnímu obrázku.

GDI+ se rovněž stará o vykreslování dalších prvků, jako jsou souřadnicové osy a vyznačené body. Pro vykreslování se používá technika dvojitého bufferu, kdy se data nekreslí přímo na obrazovku, ale do bufferu v paměti, do kterého se nejprve vykreslí všechny prvky a až po dokončení kreslení se buffer přesune na obrazovku. Tímto se eliminuje velmi nepříjemné blikání obrazovky.

Standardní kreslení pomocí GDI+ je pro některé potřeby příliš pomalé. Při kreslení čar, které se pohybují podle kurzoru by se čáry vykreslovaly se zpožděním a působilo by to rušivě. Pro tento účel jsem zde využil techniku, kdy se místo kreslení čáry invertují hodnoty barvy bodů, které mají tvořit čáru. Při posunu čáry na novou pozici není potřeba celý obrázek znovu překreslit, ale stačí u původní čáry znovu invertovat hodnoty bodů.

5.3.3 Použití dll knihovny

Použití knihovny Rdll_GraphDig.dll je velmi jednoduché. Knihovna obsahuje třídu CRGrDigDlg, která vytváří dialogové okno. Pro výměnu dat zde slouží 2 struktury. GraphInputData, která obsahuje adresu obrázku grafu a GraphOutputData, ve které je dvou dimenzionální pole obsahující přepočtené hodnoty bodů křivek. V první dimenzi jsou křivky, ve druhé jednotlivé body. Každý bod dále obsahuje X a Y souřadnici.

Ukázka použití dll knihovny:

```
GraphInputData init;
init.imagePath = imagePath;
GraphOutputData output;

CRGrDigDlg dlg;
dlg.InitiateData(init);

dlg.DoModal();

dlg.GetFinalData(output);
dlg.DestroyWindow();
```

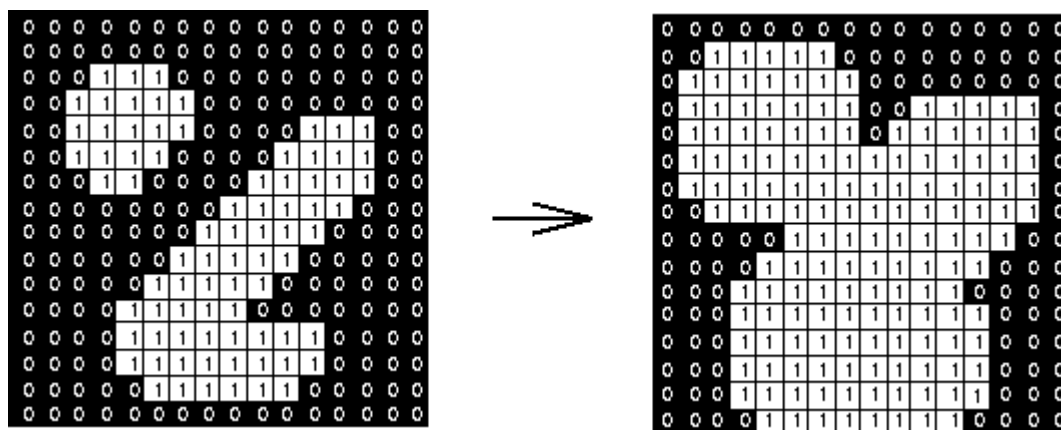
5.4 Vývoj druhé části aplikace

Ve druhém vývojovém kroku aplikace je hlavním cílem automatické nalezení křivky v obrázku grafu za použití již zmíněné knihovny pro počítačové vidění OpenCV. Veškeré funkce související s knihovnou OpenCV jsou implementovány v třídě LineTracker, která rozšíří aplikaci z první části vývoje.

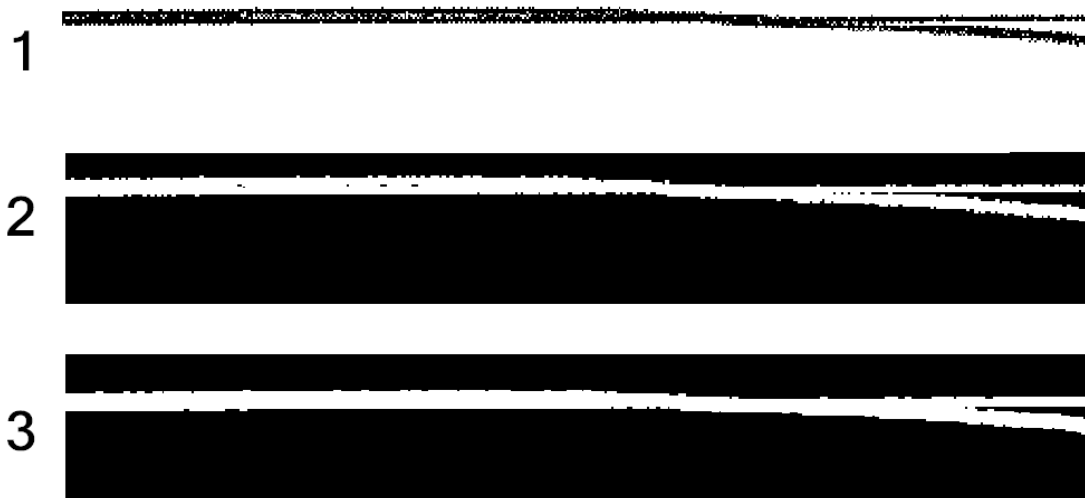
Algoritmus pro nalezení křivky je odlišný od algoritmů, které používají konkurenční programy. Většina z nich se řídí podle barvy, což je úspěšné pouze v případě barevných grafů. Mnou navržený algoritmus vyhledává trasu křivky podle spojitě oblasti v obrázku. Není závislý na barvě, tloušťce, ani tvaru křivky. Rovněž by si měl poradit s chybami v obrázku, které vzniknou například špatným naskenováním. Algoritmus pro vyhledání křivky je rozdělen do několika kroků s maximální snahou využití možností knihovny OpenCV.

5.4.1 Úprava obrázku a filtrace

Prvním krokem pro nalezení křivky je úprava obrázku s grafem. Nejprve se obrázek převede do černobílé škály. Následně se obrázek podle potřeby rozmáže nástrojem Blur, čímž se zbaví některých chyb. Dále se převede z černobílé škály do dvoubarevné škály černé – bílé. Poslední chyby v obrázku se odstraní nástrojem Dilate. Tento nástroj patří do skupiny morfologických nástrojů. Funguje na principu postupného procházení jednotlivých bodů obrázku. U každého bodu použije masku, která určí sousední body. Masku bývá nejčastěji čtverec, nebo kruh. Zde je použit čtverec o straně 3 body. Dále se zjistí, který bod obrázku z oblasti vymezené maskou má největší hodnotu. Tato hodnota se následně nastaví všem bodům pod maskou. Vizualně se tímto nástrojem zvětší světlé oblasti. Princip je znázorněn na obrázku 6. [14],[15]



Obr. 6: Princip funkce nástroje Dilate [14]

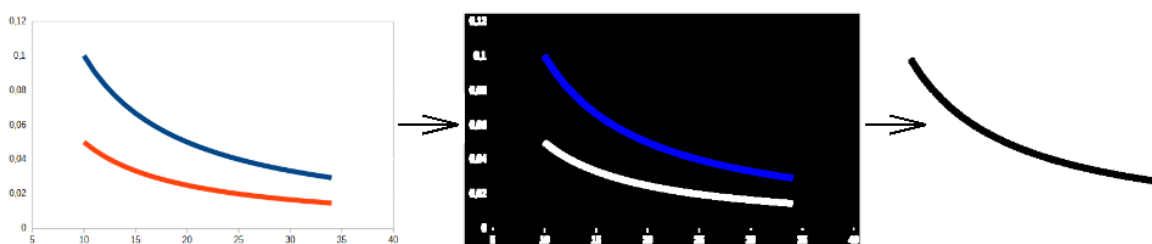


Obr. 7: Postup filtrování obrázku

Na obrázku 7 jsou zobrazeny jednotlivé kroky úpravy obrázku. Bod 1 je původní křivka. V tomto stavu obsahuje spoustu chyb a není použitelná pro další zpracování. Bod 2 je křivka upravená rozmazáním a převedená do dvoubarevné škály. V bodě 3 je finálně upravená křivka po použití nástroje Dilate. Tato křivka je již dostatečně vyčištěná a je možné ji dále zpracovat.

5.4.2 Nalezení oblasti křivky

Dalším krokem je nalezení oblasti křivky, která má být vystopována. V tomto kroku už je potřeba zapojení uživatele, který klepnutím na některou část požadované křivky určí výchozí bod. Od zvoleného bodu se nástrojem FloodFill vyplní spojitá oblast s křivkou a vše mimo tuto vyznačenou oblast se vymaže. Tímto se z obrázku odstraní všechny nepotřebné objekty.

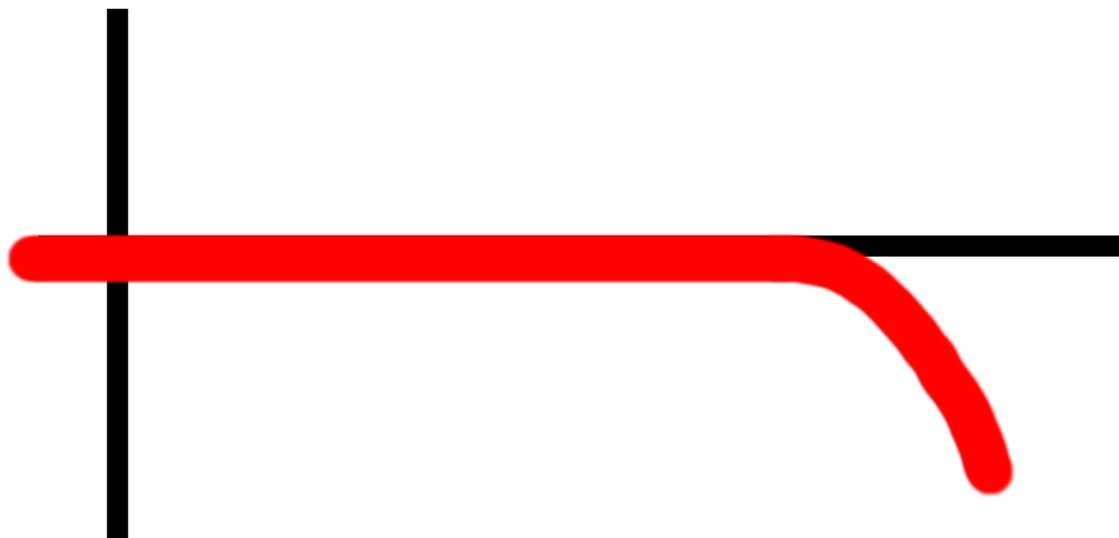


Obr. 8: Odstranění nepotřebných objektů z obrázku

5.4.3 Zmapování nalezené oblasti

V tomto kroku začíná samotná digitalizace křivky. Hledaná křivka se nachází v oblasti získané v předchozí části algoritmu. Pokud je křivka oddělená od ostatních čar jako na obrázku 8, nalezená oblast obsahuje pouze samotnou křivku. Většinou však bývá obrázek grafu komplikovanější, obsahuje více křivek, které se mohou místy protínat, případně osy a jiné čáry, které zasahují do samotné křivky. V tom případě i nalezená oblast obsahuje všechny křížící se čáry. Všechny tyto zasahující objekty mohou zkomplikovat hledání křivky. Z tohoto důvodu je potřeba ke všem čarám a křivkám v nalezené oblasti vytvořit kostru, která popíše celou nalezenou oblast a všechna křížení čar.

Pro názornou ukázkou získání kostry použijí graf na obrázku 9. Tento obrázek je pouze siluetou grafu a nepochází z žádné reálné situace. Na obrázku se nachází černě vykreslené osy a červená křivka.



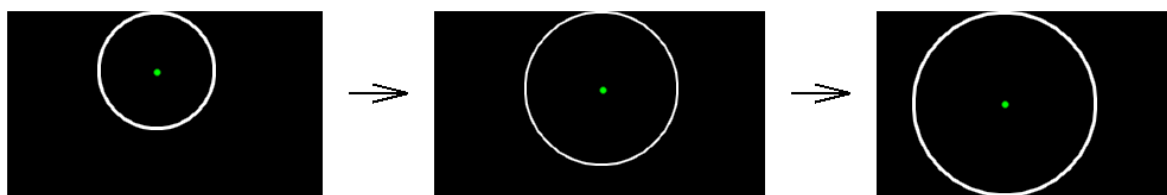
Obr. 9: Ukázkový obrázek grafu

Po úpravě a vyfiltrování se nalezne oblast křivky (obrázek 10). Vzhledem k tomu, že se všechny křivky v původním obrázku překrývají, jsou obsaženy i v nalezené oblasti křivky.



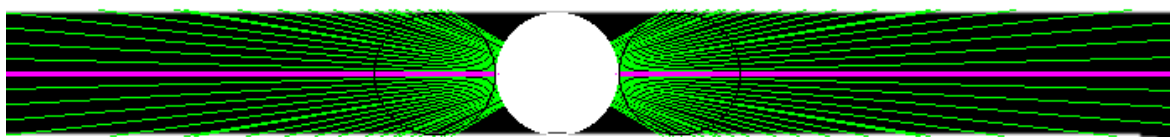
Obr. 10: Oblast křivky ukázkového grafu

Následuje výpočet prvního bodu křivky. Uživatel klepnutím na některou část křivky zadá počáteční bod. Kolem tohoto bodu se vytvoří kružnice o maximálním možném poloměru tak, aby kružnice nepřesahovala vymezenou černou oblast. Počáteční bod se poté postupně posunuje a poloměr kružnice se zvětšuje. Jakmile už není možné posunout bod a zvětšit poloměr, znamená to, že byl nalezen přesný střed křivky. Tímto postupem je možné najít střed křivky libovolného tvaru a směru. Princip je znázorněn na obrázku 11.

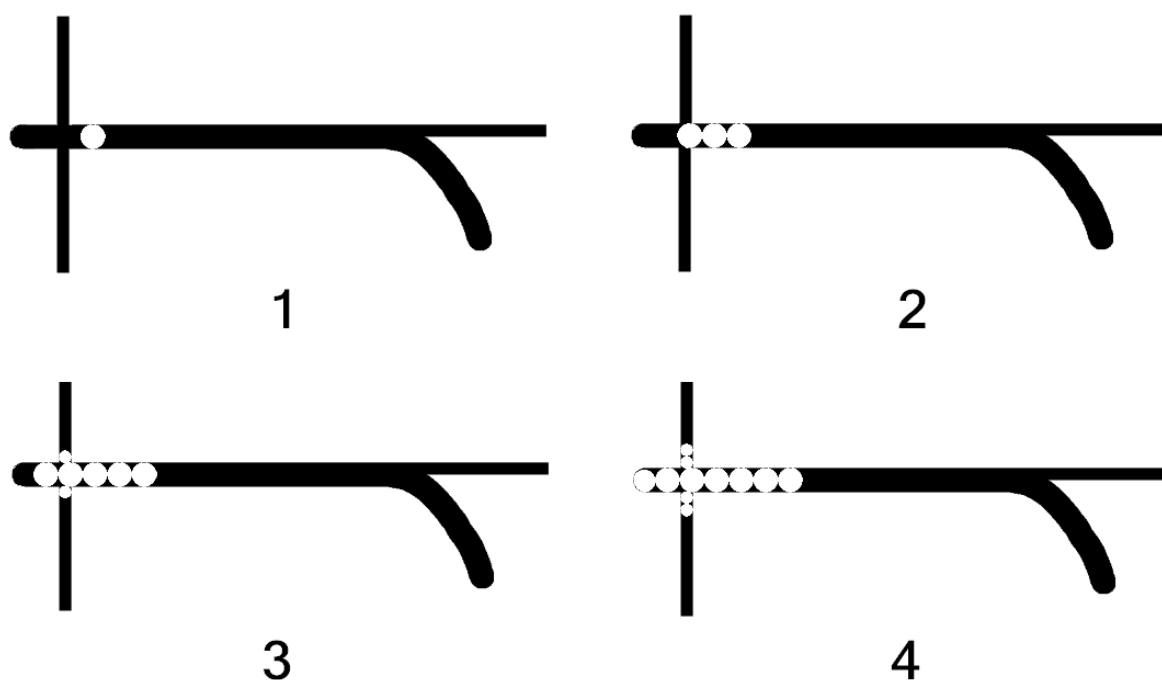


Obr. 11: Nalezení středu křivky

Po nalezení prvního bodu se prozkoumá jeho okolí. Z každého bodu tvořícího obvod kružnice se vypočítá normála (zelené čáry na obrázku 12). Čára tvořící normálu nesmí opustit černě vymezenou oblast. Podle délky jednotlivých normál se zjistí na kolika místech se kružnice dotýká okraje a kolik tedy bude mít sousedních bodů. V tomto případě to budou 2 body. Z nalezených normál se určí, které jsou nejdelší (fialové čáry na obrázku 12) a v jejich směru se budou hledat další body.

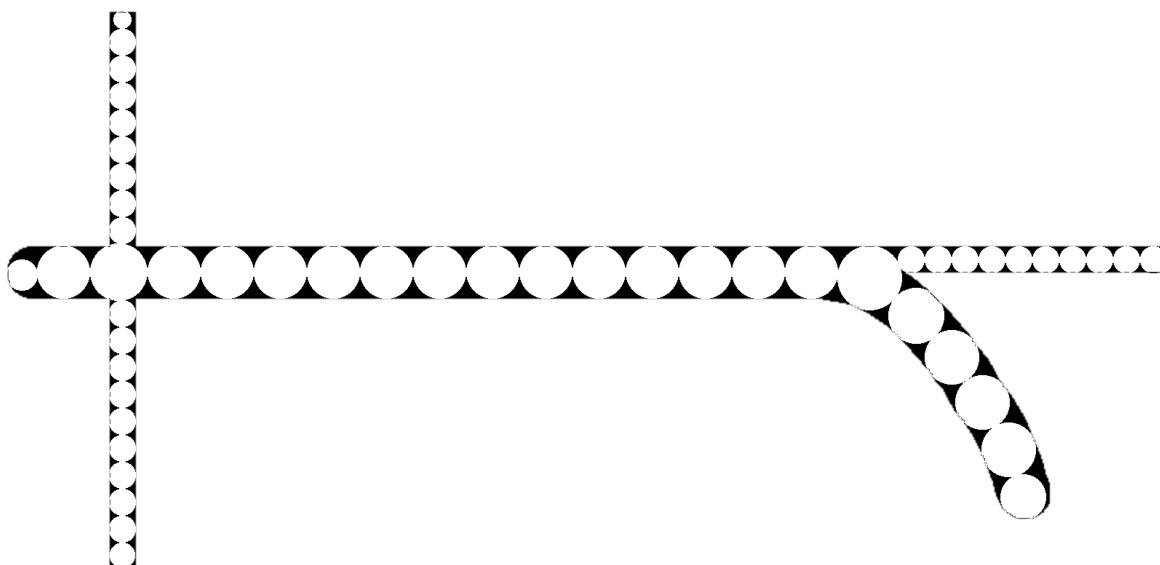


Obr. 12: Hledání směru následujících bodů



Obr. 13: Postup vytváření kostry

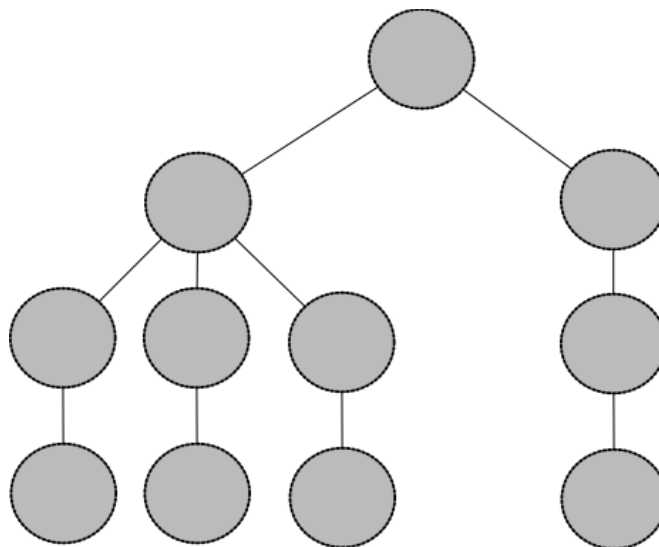
Následující body se určí stejným způsobem jako první bod. V místě, ve kterém bylo určeno pokračování křivky dalším bodem (fialová čára na obrázku 12) se sestrojí kružnice s největším možným poloměrem. Kromě podmínky nepřesáhnutí černé oblasti jsou zde další podmínky. Kružnice se musí dotýkat předchozí kružnice a zároveň nesmí protnout kružnici jiného bodu. Tento postup se rekurzivně opakuje tak dlouho, dokud se neprojde celá černá oblast. Postupné procházení je znázorněno na obrázku 13. Na obrázku 14 je kompletně zmapovaná oblast křivky.



Obr. 14: Zmapovaná oblast křivky

Po dokončení mapování je celá oblast pokryta kruhy. Středů jednotlivých kruhů pak tvoří body, jejichž spojením vznikne kostra oblasti křivky. Každý kruh má kromě svého středu definovány další vlastnosti, které později pomohou správně určit výslednou křivku. Mimo jiné jsou to poloměr kruhu a vektor směru od předchozího kruhu.

Jak je z obrázku 13 patrné, hledání kruhů začíná od prvního výchozího bodu a od něj se postupně pokračuje do všech stran. Každý kruh má referenci na předchozí kruh, ze kterého vzniknul (rodič) a na kruhy, které z něj pokračují (potomci). Takto vznikne stromová struktura, kdy výchozí bod tvoří kořen stromu a z něj se postupně rozvětvují uzly podle toho, jak se vytváří následující kruhy. Na obrázku 15 je část stromu, který vznikne z příkladu na obrázku 13.



Obr. 15: Stromová struktura kostry

5.4.4 Nalezení křivky

Z předchozího kroku dostaneme zmapovanou oblast křivky ve formě stromu. Při finálním hledání křivky se postupuje od kořene stromu směrem dolů. Tam kde se strom rozvětjuje je potřeba určit kterou větví se má pokračovat. K tomu slouží několik kritérií. Hlavním kritériem je změna směru křivky. Nejprve se tedy vyřadí větve, u kterých se vektor směru příliš odlišuje od vektoru směru předchozího bodu a křivka by tak prudce zabočila. U zbylých větví se porovná jejich délka a zvolí se nejdelší z nich.

U složitějších grafů se může stát, že se v některém kroku nezvolí správná větev a nenajde se tak správná křivka. Pro tyto případy může uživatel kliknutím na další část křivky, která nebyla nalezena, určit místo, kterým má křivka procházet. Od tohoto bodu, kde uživatel kliknul, se zjistí nejbližší uzel stromové struktury. Tento uzel se označí jako místo, kterým křivka musí projít. Od tohoto uzlu se postupuje zpátky ke kořenu stromu a každému uzlu po cestě se nastaví značka určující správnou cestu. Následně se znovu provede detekce křivky. Postupuje se zase od kořenu stromu tak jako při prvním hledání. V místech, kde se strom rozvětjuje se nejprve zjistí, jestli některá z větví neobsahuje značku označující cestu k uzlu, kde uživatel kliknul. Pokud ano, pokračuje se automaticky touto větví a na ostatní kritéria se nebere ohled. Uživatel může takto označit i další body, pokud ani na druhý pokus nebyla nalezena celá křivka.

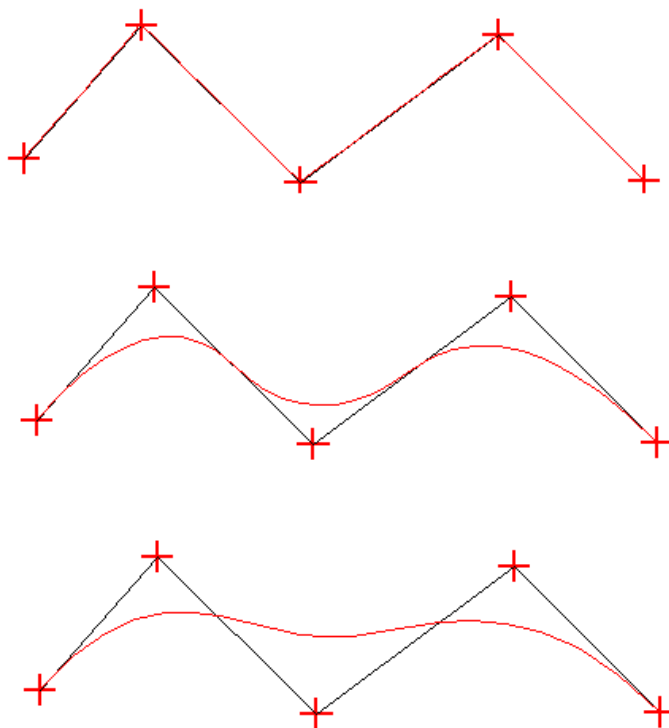
5.4.5 B-Spline křivky

Ručním, nebo automatickým režimem hledání křivky získáme několik bodů ležících na křivce. Pro popis celého průběhu křivky je to málo. Proto byla jedním z požadavků možnost proložení bodů B-Spline křivkou se zvoleným stupněm.

B-Spline křivka patří mezi polynomiální křivky. Její tvar určuje řídící polygon a stupeň. Pro vykreslení křivky jsem využil de Boorův algoritmus. Jedná se o zobecněný de Casteljau algoritmus pro vykreslování Beziérových křivek. Základem je výpočet vektoru uzlů, který je dán počtem bodů řídícího polynomu a požadovaným stupněm křivky. Pro 6 řídících bodů a druhý stupeň křivky vypadá následovně: [0, 0, 0, 1, 2, 3, 4, 4, 4]. Dále se použije rekursivní vztah (vzorec 1), kde \mathbf{u} je vektor uzlů, i je index vrcholu řídícího polynomu, k je požadovaný stupeň křivky a t je parametr výpočtu.

$$N_i^k(u) = \frac{t - u_i}{u_{i+k} - u_i} \cdot N_i^{k-1}(u) + \frac{u_{i+k+1} - t}{u_{i+k+1} - u_{i+1}} \cdot N_{i+1}^{k-1}(u) \quad (1)$$

Tímto vzorcem se vypočítá hodnota de Boorovy funkce pro vykreslení bodů B-Spline křivky v rozsahu parametru $t \in [u_0, u_m]$; $m = n + k + 1$, kde n je počet bodů řídícího polygonu. [16] [17]



Obr. 16: B-Spline křivky

Na obrázku 16 je ukázka B-Spline křivky prvního, druhého a třetího stupně. Červenými křížky jsou označeny řídicí body, které spolu s černými čarami tvoří řídicí polygon.

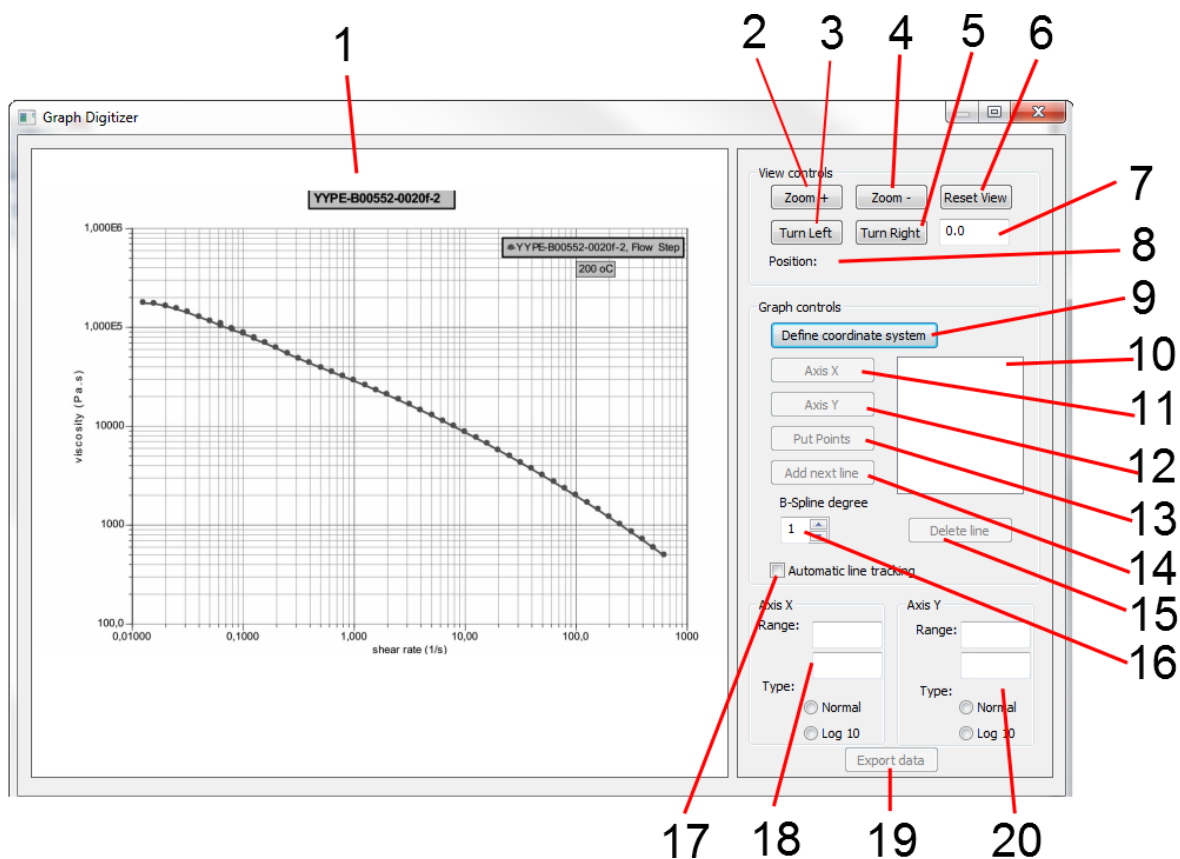
5.5 Ovládání aplikace

Aplikaci tvoří dll soubor, který není samostatně spustitelný. Spustit se dá pouze z kódu jiného programu. Pro spouštění slouží soubor Digitizer.exe, který umožňuje otevřít obrázek grafu, spustit dialogové okno z dll souboru a na konci uložit získané hodnoty do textového souboru.



Obr. 17: Spouštěcí okno digitalizéru

Po spuštění EXE souboru se objeví spouštěcí okno aplikace (obrázek 17). Jsou zde pouze 2 volby. Stisknutím tlačítka *Load Image* se otevře standardní Windows okno s výběrem souboru. Po potvrzení vybraného souboru, za předpokladu že není obrázek poškozený, se spustí hlavní okno aplikace. V opačném případě se zobrazí chybová hláška a je nutné zvolit jiný soubor. Podporované formáty obrázků jsou JPG, BMP, PNG a TIFF.



Obr. 18: Hlavní okno aplikace

Po úspěšném načtení obrázku se zobrazí hlavní okno aplikace (obrázek 18).

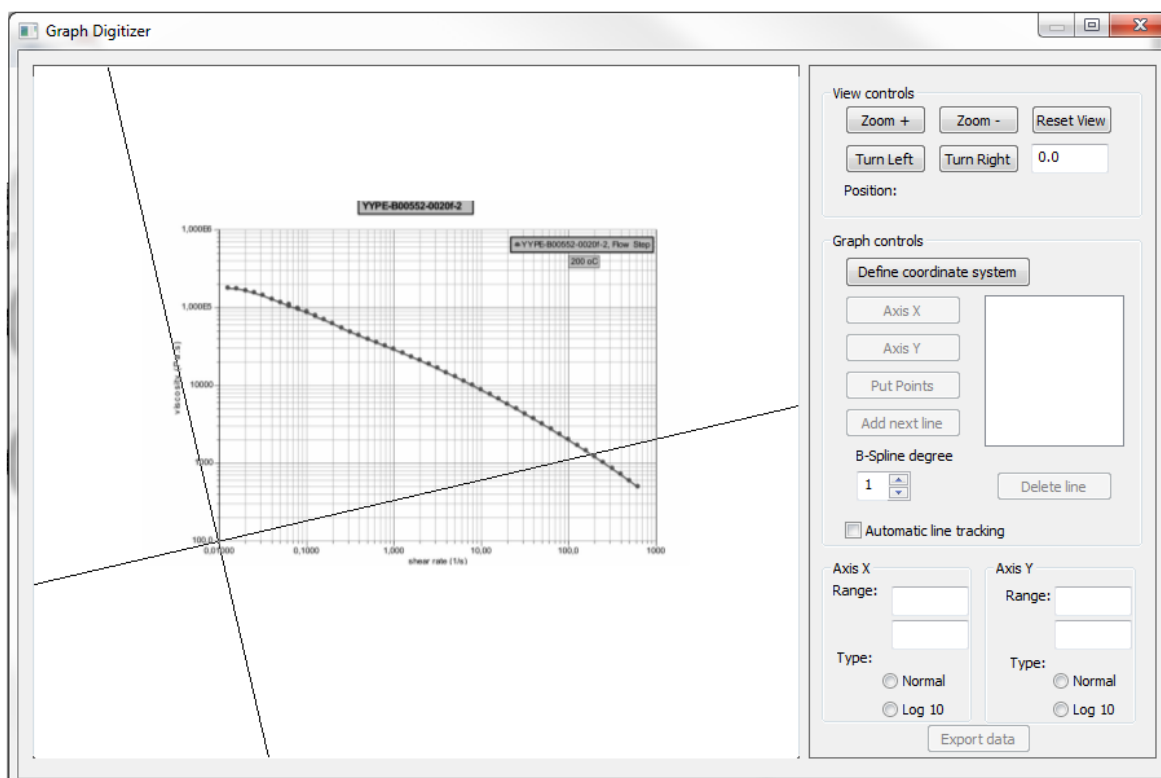
Popis ovládacích prvků:

1. Okno s obrázkem grafu.
2. Přiblížení obrázku.
3. Otočení obrázku proti směru hodinových ručiček.
4. Oddálení obrázku.
5. Otočení obrázku po směru hodinových ručiček.
6. Vrácení obrázku do původní velikosti a natočení.
7. Otočení obrázku o zvolený úhel.
8. Zobrazení X a Y pozice kurzoru (pouze pokud jsou umístěny souřadnicové osy a nastaven rozsah)
9. Umístění souřadnicového systému.
10. Okno s digitalizovanými křivkami.
11. Umístění osy X.

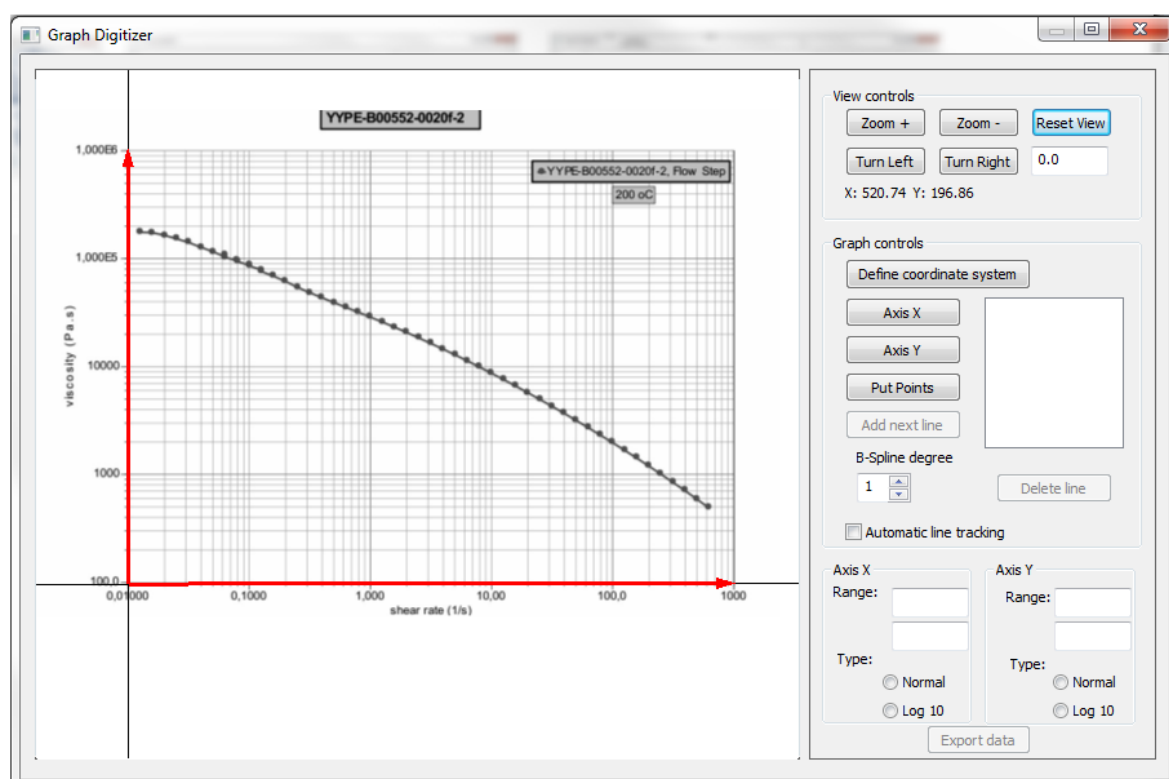
12. Umístění osy Y.
13. Přepnutí do módu přidávání bodů.
14. Přidání další křivky.
15. Smazání křivky.
16. Volba stupně pro aproximaci B-Spline křivkou.
17. Povolení automatické detekce křivky.
18. Rozsah a typ osy X.
19. Exportování digitalizovaných dat.
20. Rozsah a typ osy Y.

V okně s obrázkem grafu je možné provádět transformační úpravy obrázku. Pohybem myši při současném držení levého tlačítka lze s obrázkem pohybovat do všech směrů. Otáčením kolečkem na myši se obrázek přibližuje a oddaluje vzhledem k umístění kurzoru. Transformace je možné provádět i pomocí tlačítek v pravém menu aplikace pod kolonkou *View controls*. Zde jsou tlačítka pro přibližování a oddalování (*Zoom +*, *Zoom -*). Dále je možné dvojicí tlačítek *Turn Left* a *Turn Right* obrázkem otáčet po 5° krocích. Pro otočení obrázku o přesný úhel je možné zapsat číselnou hodnotu úhlu do kolonky vedle těchto tlačítek. Pro zrušení všech transformací slouží tlačítko *Reset View*.

Prvním krokem pro digitalizaci křivek je umístění kříže určujícího osy souřadnicového systému. Tato akce se aktivuje stisknutím tlačítka *Define Coordinate System*. Následně se kliknutím do obrázku grafu určí místo nulového bodu, neboli místa kde se protínají osy X a Y. Po umístění tohoto bodu se na obrazovku vykreslí kříž, který se pohybuje spolu s kurzorem myši (obrázek 19). Je potřeba jej zarovnat s osami v obrázku grafu a potvrdit kliknutím. Po potvrzení se aktivuje zadávání délky osy X. Tahem myši se vybere směr a délka osy a potvrdí stisknutím levého tlačítka myši. Následně se stejným způsobem umístí osa Y. Tu je možné umístit pouze kolmo na osu X. Ukázka správného umístění os je na obrázku 20. Pokud se nepodařilo některou osu umístit správně, je možné její umístění změnit. K tomu slouží tlačítka v pravém menu *Axis X* a *Axis Y*. Každý z těchto kroků lze navíc kdykoliv přerušit stisknutím klávesy *Esc* a pokud ještě nebyla potvrzena nová pozice, vrátí se do původního stavu.

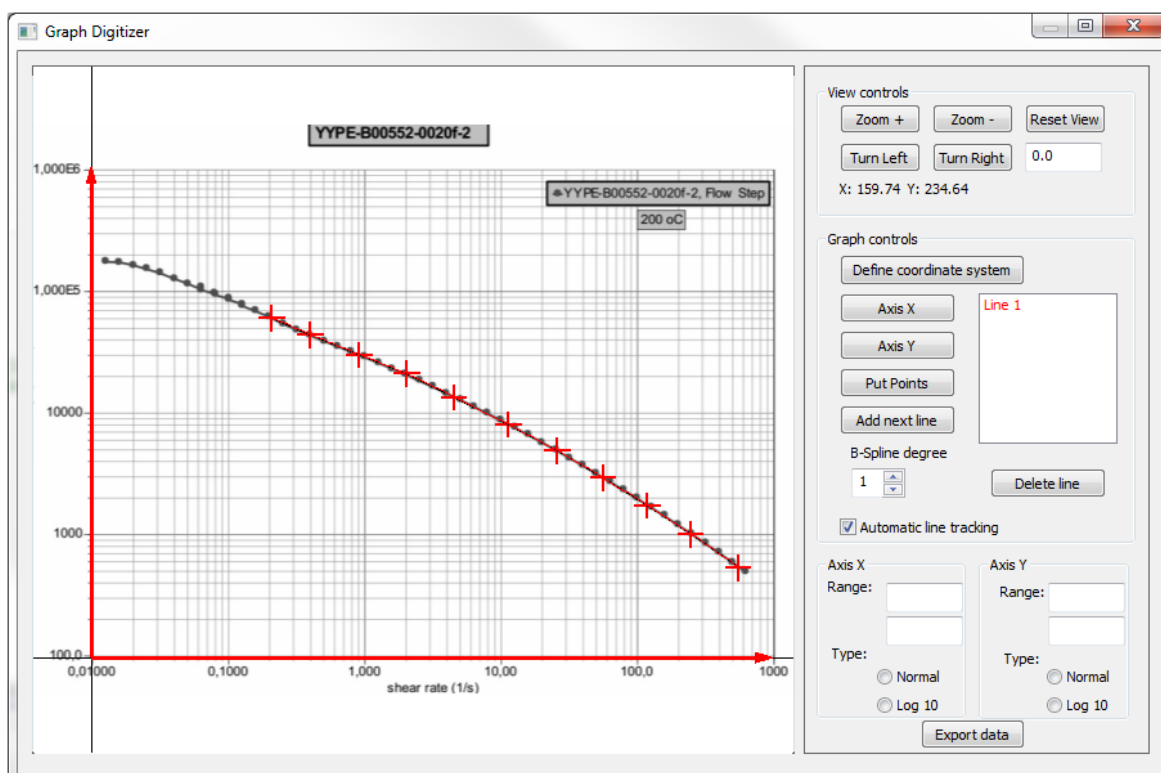


Obr. 19: Umístění souřadnicového kříže



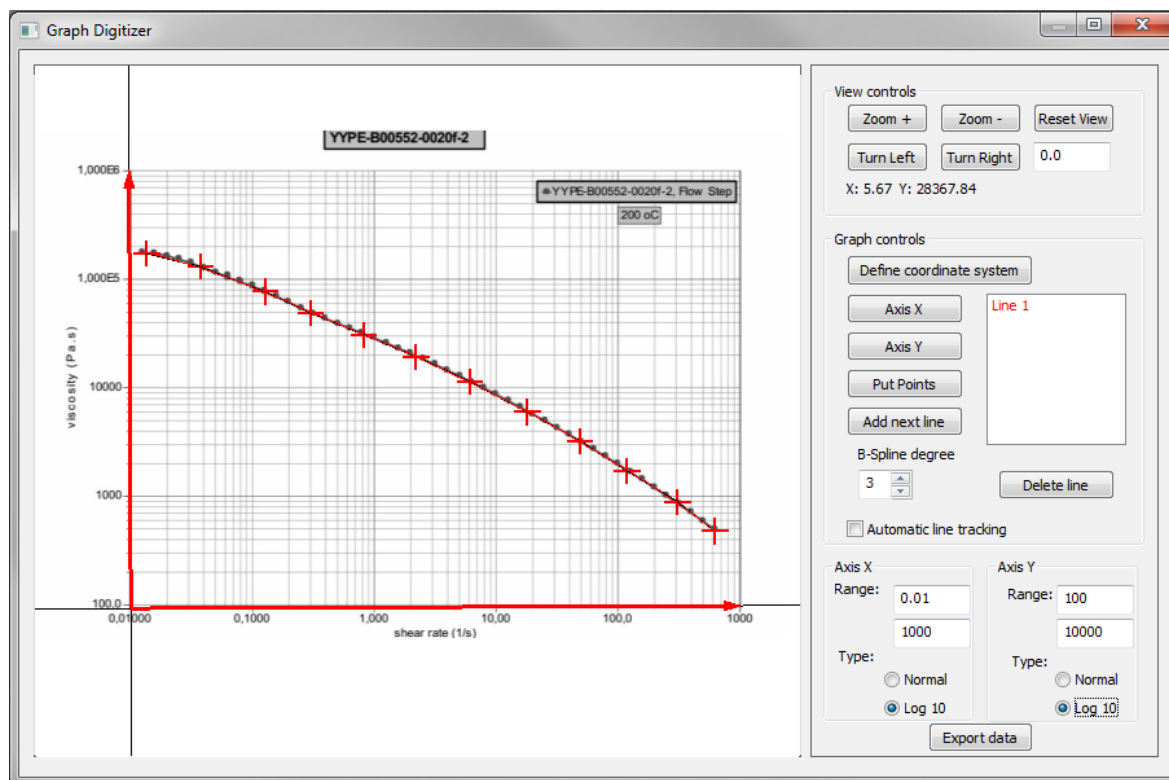
Obr. 20: Umístění os

Následuje vyznačení bodů na křivce. Stisknutím tlačítka *Put Points* se aktivuje ruční zadávání bodů. Body se umísťují kliknutím do obrázku grafu. Pokud je v grafu spojitá křivka, je možné využít automatické detekce křivky. Pro aktivaci je potřeba zaškrtnout políčko *Automatic line tracking* a poté zvolit možnost *Put Points*. Následně stačí kliknout na vybranou křivku v obrázku grafu a dojde k její automatické detekci a vykreslení do obrázku. Pokud je graf členitější a není detekována celá křivka (obrázek 21), nebo pokud se křivka kříží s jinou čarou, kterou algoritmus špatně detekoval jako pokračování křivky, stačí kliknout na místo na křivce, které nebylo detekováno a algoritmus pro detekování křivky se pokusí vést křivku skrz tento bod. Správně detekovaná křivka je na obrázku 22. Pokud by i přes to nebyla křivka nalezena celá, lze body doplnit ručně. Stačí odškrtnout volbu *Automatic line tracking* a kliknutím vložit body na místa, která algoritmus nenašel.

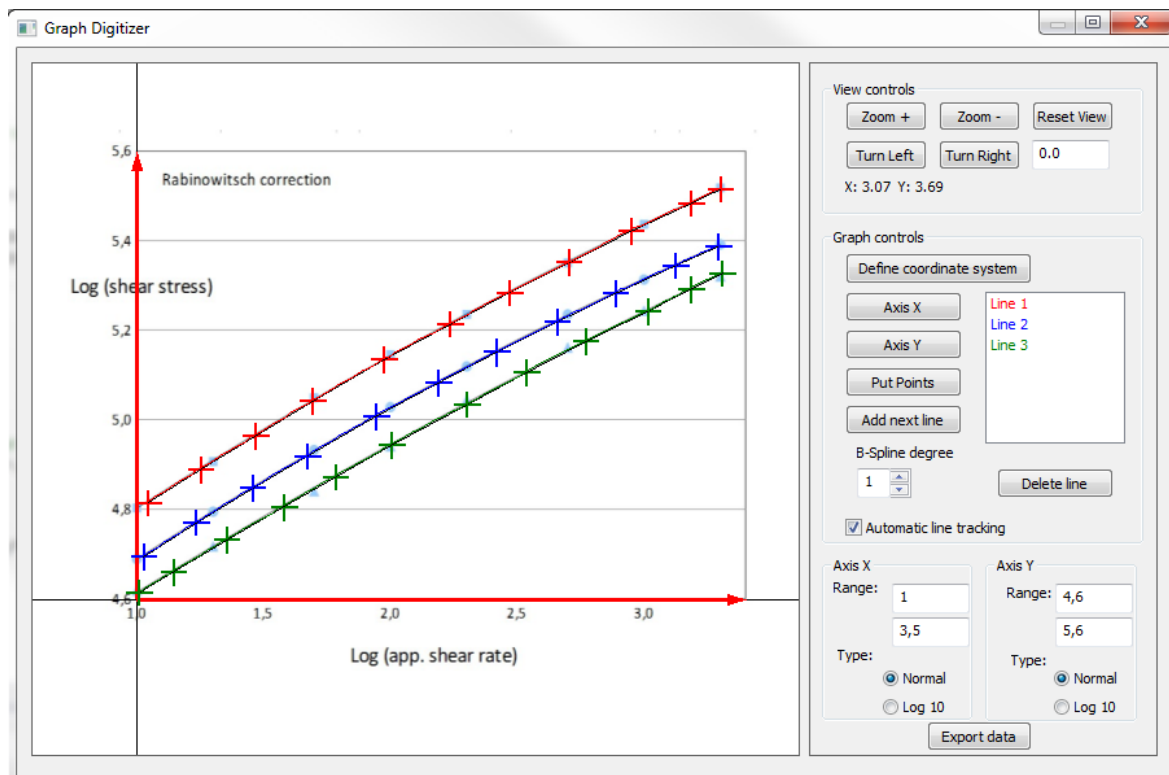


Obr. 21: Neúplná detekce křivky

Pokud se v obrázku vyskytuje více křivek, pracuje se s nimi obdobně jako s jednou křivkou. Po vyznačení první křivky se stisknutím tlačítka *Add next line* přidá další křivka a pokračuje se vyznačováním druhé křivky. Jednotlivé křivky jsou od sebe barevně odděleny. Jejich seznam se nachází v pravé části menu, kde je možné celou křivku smazat jejím označením v okně a stisknutím tlačítka *Delete line*. Ukázka grafu se třemi křivkami je na obrázku 23.



Obr. 22: Správně detekovaná křivka



Obr. 23: Graf s více křivkami

Pro lepší pokrytí tvaru křivky je možné ji aproximovat kvadratickou (2. řádu) B-Spline křivkou, nebo kubickou (3. řádu) B-Spline křivkou zvolením příslušného stupně v kolonce *B-Spline degree*. Jak vypadá aproximace B-Spline křivkou je demonstrováno na obrázku 16 v kapitole 5.4.5.

Vyznačené body na křivce lze také editovat. Pro přesunutí bodu stačí kliknout levým tlačítkem na požadovaný bod. Barva bodu se změní na inverzní a při držení stisknutého levého tlačítka myši lze s bodem pohybovat. Mazat body je možné dvěma způsoby. Buď označením požadovaného bodu kliknutím levým tlačítkem myši a následně stisknutím klávesy *Delete*, nebo pro mazání více bodů najednou nejprve označit skupinu bodů podržením pravého tlačítka myši, kterým se vykreslí výběrový obdélník. Vybrané body změní barvu a smažou se stisknutím klávesy *Delete*. Označení bodů je možné zrušit stisknutím klávesy *Esc*. Pokud graf obsahuje více křivek, je nutné pro editaci, nebo mazání vybrat, která křivka má být upravena. To se určí vybráním požadované křivky kliknutím na její název v pravém okně v menu *Graph controls*.

Posledním krokem digitalizace je uložení výsledků. Aby mohly být vyznačené body přepočítány, je potřeba nastavit rozsah os, které se v předchozích krocích vyznačily pomocí červených šipek. Hodnoty se nastavují do políček *Range* pro každou osu zvlášť. Do horního políčka se nastaví hodnota, která odpovídá místu, kde červená šipka začíná. Do spodního políčka se nastaví hodnota odpovídající konci šipky. Pro každou osu se také zvolí jestli má lineární, nebo logaritmický průběh. Následně se stisknutím tlačítka *Export data* zavře hlavní okno digitalizéru a místo něj se otevře dialog pro uložení textového souboru s přepočítanými hodnotami. Po zadání umístění a názvu souboru se do něj uloží přepočítané hodnoty pro všechny křivky. V případě, že se z nějakého důvodu nepodaří data uložit na první pokus, stisknutím tlačítka *Save Data* je možné proces ukládání zopakovat.

5.6 Porovnání s konkurenčními nástroji

Pro zjištění kvality vytvořeného programu jsem jej porovnal s podobnými programy představenými v kapitole 2. Jelikož se programy funkčně odlišují, otestoval jsem je pouze na schopnost automatické detekce křivek. Tu umožňují programy Curvesnap V1.0, Graph Digitizer Scout 1.2.4 a Digitize It 2.0. Každý z těchto programů jsem otestoval na deseti zkušebních grafech, které jsou v příloze 1 této práce.

5.6.1 Testování programu Curvesnap V1.0

Tento program má ze všech testovaných programů nejhorší auto detekční schopnost. Hlavním nástrojem detekce je hledání křivky podle barvy. Je to velmi primitivní nástroj, který vyžaduje pro správnou detekci unikátní barvu křivky a ve většině grafů je nepoužitelný. Druhou možností je hledání podle spojitě oblasti. Tato funkce vyžaduje naprostou izolaci křivky od ostatních čar a objektů. V žádném ze vzorových grafů se tímto nástrojem nepodařilo křivku získat. Stejně tak posledním nástrojem, výběrem křivky výběrovým obdélníkem se nepodařilo získat správnou křivku. Program sice umožňuje upravit obrázek pomocí tužky a gummy, ale tato úprava by byla časově náročnější, než vyznačení bodů ručně.

5.6.2 Testování programu Graph Digitizer Scout 1.2.4

Stejně jako předchozí program, Graph Digitizer Scout umožňuje hledání křivky na základě její barvy. U grafů, které obsahují křivky s unikátními barvami dokázal najít požadovanou křivku bez problémů. U jednobarevných grafů se ukázal jako absolutně nepoužitelný. Kromě detekce podle barvy umožňuje i detekci podle světlosti. Použití je však komplikované a u žádného ze zkušebních grafů se mi tímto nástrojem nepodařilo křivku najít.

5.6.3 Testování programu Digitize It 2.0

Tento program je z testovaných programů třetích stran nejlepší v hledání křivek. Dokáže si poradit i s některými jednobarevnými křivkami. Zároveň jako jediný umožňuje digitalizování nespojitých grafů. Vyžaduje to však velmi zdoluhavé nastavování citlivosti a u testovaného nespojitého grafu by bylo daleko rychlejší vyznačit body ručně. I přes tyto vlastnosti si s polovinou testovaných křivek nedokázal poradit.

5.6.4 Testování vlastního programu Compuplast Digitizer

Autodetekční schopnost tohoto programu byla zaměřena na hledání křivky podle jejího tvaru a ne podle barvy. Program jasně vyniká v digitalizování jednobarevných grafů a dokáže si poradit i s četným křížením čar. Problémy s detekcí křivky nastávají v případech, kdy je v grafu několik křivek velmi blízko u sebe. Algoritmus pro stopování tvaru křivky je nastaven velmi volně a křivky blízko sebe bere jako jednu poškozenou křivku. Na druhou stranu si dokáže poradit i s nestandardními křivkami, ve kterých se vyskytují různé symboly a křížení.

Tab. 5. Porovnání automatické detekce

Obrázek grafu	Curvesnap V1.0	Graph Digitizer Scout 1.2.4	Digitize It 2.0	Compuplast Digitizer
1	✓	✓	!	!
2	✗	✗	✓	✓
3	✗	✗	✗	!
4	!	✓	✓	!
5	✗	✗	✓	✗
6	✗	✗	✗	✓
7	✓	✓	!	!
8	✗	✗	✗	✓
9	✗	✗	✗	✓
10	!	✗	✗	!

V tabulce 5 je porovnání schopností automatické detekce křivky všech testovaných programů na všech deseti vzorových grafech.

Význam symbolů použitých v tabulce:



Všechny křivky byly správně detekovány.



Některé křivky nebyly detekovány přesně.



Některá z křivek nebyla detekována správně, nebo jsou křivky detekovány velmi nepřesně.



Žádnou z křivek se nepodařilo detekovat.

ZÁVĚR

Většina vyzkoušených nástrojů pro hledání trasy křivky v bitmapovém obrázku funguje na principu hledání podle spojitých barevných oblastí. Tento postup je úspěšný pouze u grafů s barevnými křivkami. U jednobarevných grafů jsou nepoužitelné. Z hlediska ovládání mají všechny testované programy zbytečně složité a zdlouhavé nastavování souřadnicového systému a velmi omezené možnosti manipulace s obrázkem. Proto jsem se na tyto nedostatky zaměřil a pokusil se navrhnout vlastní aplikaci tak, aby umožňovala co možná nej pohodlnější a nejrychlejší ovládání. Obrázek grafu není potřeba před digitalizací ořezávat, nebo jinak upravovat externími nástroji. Všechny potřebné transformace je možné provést přímo v programu. Pro automatické hledání křivky jsem použil odlišný postup, než testované aplikace. Křivka se nehledá podle barvy, ale podle spojité oblasti, která má odlišnou barvu od podkladu. Tento algoritmus má při hledání křivek daleko větší úspěšnost a bez problému zvládá i jednobarevné grafy. Ve fázi předzpracování obrázku dokáže opravit a zpracovat i křivky poškozené například nekvalitním kopírováním. Tato oprava se ale zároveň negativně projevuje u křivek, které jsou velmi blízko sebe, jelikož je velmi těžké určit, jestli se jedná o více křivek, nebo o jednu poškozenou křivku. Pro tyto situace se mi nepodařilo najít optimální nastavení algoritmu. Další nevýhodou je výpočetní náročnost. Vyhledávání podle barvy je jednoduché a velmi rychlé. Mnou použitý způsob je naopak velmi komplikovaný. Pokud graf obsahuje velké množství čar, například hustou mřížku, musí se nejprve vytvořit mapa obsahující všechny čáry křížící požadovanou křivku. V některých případech tento výpočet trvá i několik vteřin. Pro uživatele to může být nepohodlné a pravděpodobně bude potřeba před ostrým nasazením aplikace zdrojový kód lépe optimalizovat.

I přes zmíněné nedostatky je funkčnost z mého pohledu na dobré úrovni. Na aplikaci mám v úmyslu dále pracovat a její funkce lépe propojit s programem VEL. Zkušenosti získané při vývoji algoritmu pro sledování čáry využiji při práci na dalším projektu zaměřeném na hledání kostry 2D objektů.

SEZNAM POUŽITÉ LITERATURY

- [1] COMPUPLAST® Product: Virtual Extrusion Laboratory. [online]. [cit. 2015-04-15]. Dostupné z: <http://www.compuplast.com/product>
- [2] OpenCV: Platforms. [online]. [cit. 2015-04-19]. Dostupné z: <http://opencv.org/platforms.html>
- [3] BRADSKI, Gary R. *Learning OpenCV*. Sebastopol: O'Reilly, c2008, xvii, 555 s. ISBN 978-0-596-51613-0.
- [4] Plot Digitizer. *University of South Alabama* [online]. [cit. 2015-04-10]. Dostupné z: <http://www.southalabama.edu/physics/software/plotdigitizer.htm>
- [5] Plot Digitizer. 2014. *SourceForge* [online]. [cit. 2015-04-10]. Dostupné z: <http://sourceforge.net/projects/plotdigitizer/>
- [6] CurveSnap. 2015. *Softonic* [online]. [cit. 2015-04-11]. Dostupné z: <http://curvesnap.en.softonic.com/>
- [7] VLFeat. 2015. *VLFeat.org* [online]. [cit. 2015-04-15]. Dostupné z: <http://www.vlfeat.org/index.html>
- [8] *DigitizeIt* [online]. 2015. BORMANN,. [cit. 2015-04-17]. Dostupné z: <http://digitize-it.de/>
- [9] *OpenCV: About* [online]. 2015. [cit. 2015-04-17]. Dostupné z: <http://opencv.org/about.html>
- [10] BoofCV. 2014. *BoofCV* [online]. [cit. 2015-04-17]. Dostupné z: <http://boofcv.org/>
- [11] BoofCV. *GitHub* [online]. [cit. 2015-04-17]. Dostupné z: <https://github.com/lessthanoptimal/BoofCV>
- [12] BoofCV. 2011. *SourceForge* [online]. [cit. 2015-05-5]. Dostupné z: <http://sourceforge.net/projects/boofcv/>
- [13] Windows GDI. *Microsoft: Windows Dev Center* [online]. [cit. 2015-05-5]. Dostupné z: <https://msdn.microsoft.com/en-us/library/windows/desktop/dd145203%28v=vs.85%29.aspx>
- [14] Tutorials: Noise Reduction. *Robotix* [online]. [cit. 2015-05-5]. Dostupné z: http://robotix.in/tutorials/category/opencv/noise_reduction
- [15] Tutorials: Eroding and Dilating. 2014. *OpenCV* [online]. [cit. 2015-05-8]. Dostupné z: http://docs.opencv.org/doc/tutorials/imgproc/erosion_dilatation/erosion_dilatation.html

- [16] GALLIER, Jean. 2000. *Curves and surfaces in geometric modeling: theory and algorithms*. 1st ed. San Francisco: Morgan Kaufmann, 491 s. ISBN 15-586-0599-1.
- [17] Geometrie: B-spline křivka. 2011. *Wikiknihy* [online]. [cit. 2015-05-8]. Dostupné z: http://cs.wikibooks.org/wiki/Geometrie/B%E2%80%93spline_k%C5%99ivka
- [18] Aforge.NET. 2012. *Framework* [online]. [cit. 2015-05-10]. Dostupné z: <http://www.afor-genet.com/framework/>
- [19] Aforge.NET. 2012. *Projects* [online]. [cit. 2015-05-10]. Dostupné z: <http://www.afor-genet.com/framework/projects.html>
- [20] TUCKER, Ch. 1989. *Fundamentals of Computer Modeling for Polymer Processing*. New York: Hanser Publishers, 623 s. ISBN 01-952-0766-1.
- [21] AGASSANT, J. 1991. *Polymer processing: principles and modeling*. New York: Distributed in the U.S.A. and Canada by Oxford University Press, xxiv, 475 p. ISBN 0195208641X.
- [22] TADMOR, Zehev. 2006. *Principles of polymer processing*. 2nd ed. Hoboken: Wiley, xvi, 961 s. ISBN 04-713-8770-3.
- [23] Graph Digitizer Scout. 2015. *Bytescout* [online]. [cit. 2015-05-11]. Dostupné z: <http://bytescout.com/?q=/products/enduser/graphdigitizerscout/graphdigitizerscout.html>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

BMP	Bitmap
CSV	Comma separated values
DLL	Dynamic-link library
EXE	Executable
GDI	Graphical Device Interface
MFC	Microsoft Foundation Classes
MSER	Maximally stable extremal regions
PNG	Portable Network Graphics
SIFT	Scale invariant feature transform
TIFF	Tag Image File Format.
VEL	Virtual Extrusion Laboratory

SEZNAM OBRÁZKŮ

Obr. 1: Plot digitizer 2.0.....	12
Obr. 2: CurveSnap V1.0.....	13
Obr. 3: Graph Digitizer Scout 1.2.4.....	14
Obr. 4: DigitizeIt.....	16
Obr. 5: Spojení dvou snímků na základě výpočtu klíčových bodů algoritmem SIFT [7]....	19
Obr. 6: Princip funkce nástroje Dilate [14].....	26
Obr. 7: Postup filtrování obrázku.....	26
Obr. 8: Odstranění nepotřebných objektů z obrázku.....	27
Obr. 9: Ukázkový obrázek grafu.....	28
Obr. 10: Oblast křivky ukázkového grafu.....	28
Obr. 11: Nalezení středu křivky.....	29
Obr. 12: Hledání směru následujících bodů.....	29
Obr. 13: Postup vytváření kostry.....	29
Obr. 14: Zmapovaná oblast křivky.....	30
Obr. 15: Stromová struktura kostry.....	31
Obr. 16: B-Spline křivky.....	32
Obr. 17: Spouštěcí okno digitalizéru.....	33
Obr. 18: Hlavní okno aplikace.....	34
Obr. 19: Umístění souřadnicového kříže.....	36
Obr. 20: Umístění os.....	36
Obr. 21: Neúplná detekce křivky.....	37
Obr. 22: Správně detekovaná křivka.....	38
Obr. 23: Graf s více křivkami.....	38
Obr. 24: Graf 1 – materiál Akulon® F136-C.....	49
Obr. 25: Graf 2 – materiál Petkim YYPE.....	49
Obr. 26: Graf 3 – materiál FP201B.....	50
Obr. 27: Graf 4 – materiál ME3440.....	50
Obr. 28: Graf 5 – materiál TP12962.....	51
Obr. 29: Graf 6 – materiál TP12962.....	51
Obr. 30: Graf 7 – materiál NF498E.....	52
Obr. 31: Graf 8 [20].....	53
Obr. 32: Graf 9 [21].....	53

Obr. 33: Graf 10 [22].....	54
----------------------------	----

SEZNAM TABULEK

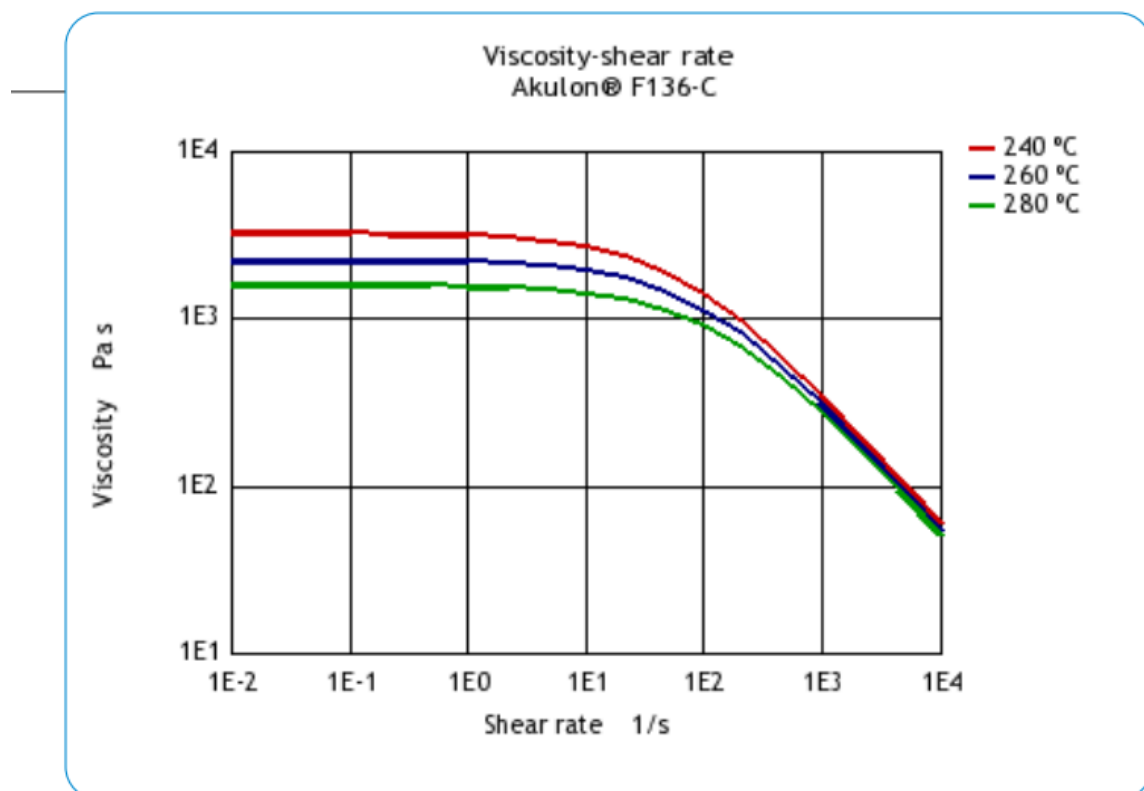
Tab. 1. Popis částí knihovny VLFeat [7].....	19
Tab. 2. Popis modulů.....	20
Tab. 3. Popis knihoven frameworku Aforge.NET [18].....	21
Tab. 4. Popis tříd DLL knihovny.....	24
Tab. 5. Porovnání automatické detekce.....	41

SEZNAM PŘÍLOH

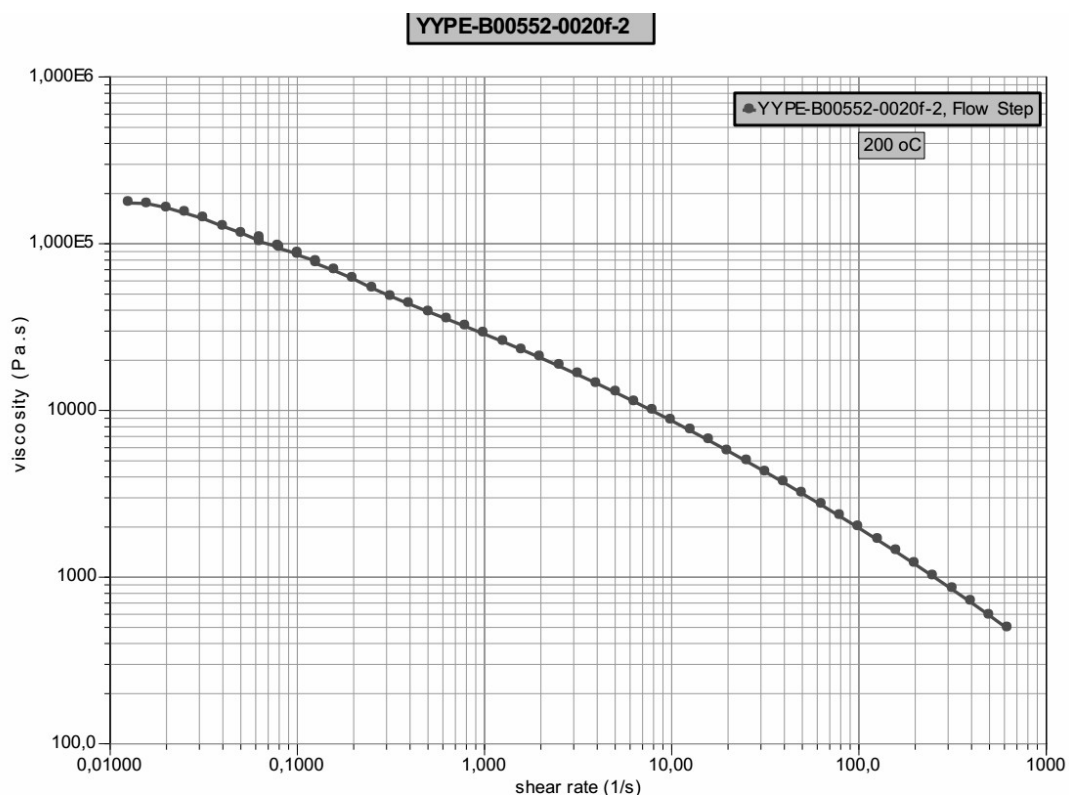
Příloha P 1: Vzorové grafy

Příloha P 2: DVD s elektronickou verzí této práce, zkompilovanou aplikací a zdrojovými kódy

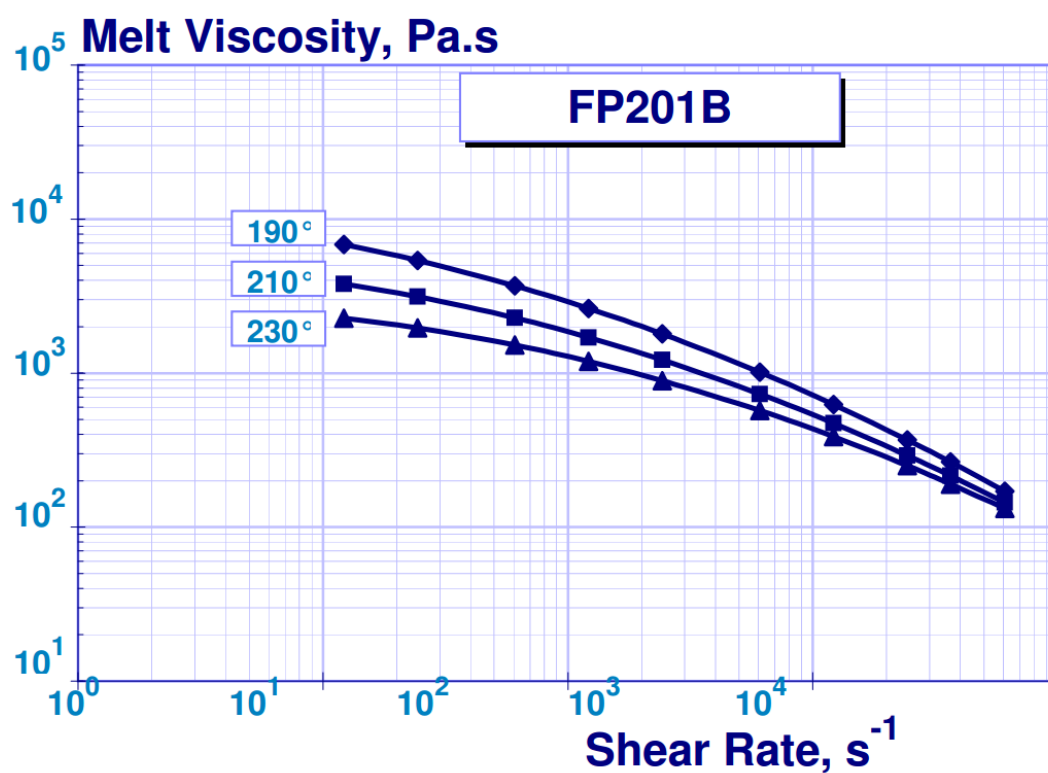
PŘÍLOHA P 1: VZOROVÉ GRAFY



Obr. 24: Graf 1 – materiál Akulon® F136-C

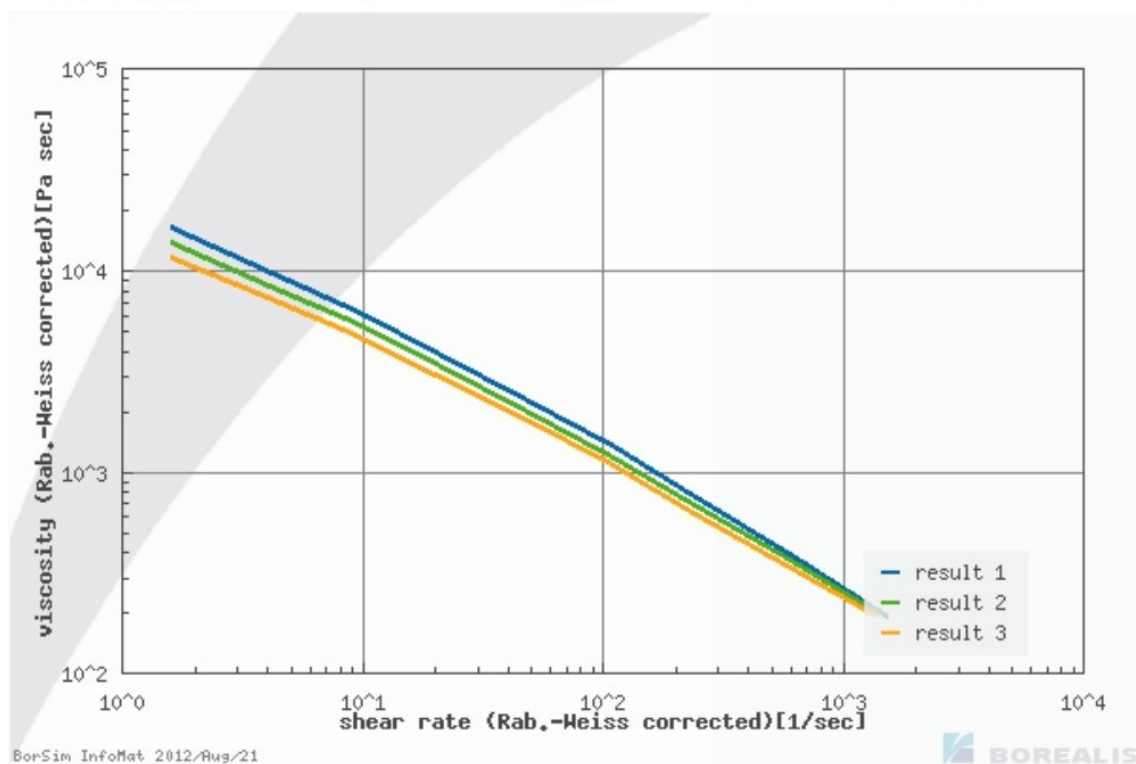


Obr. 25: Graf 2 – materiál Petkim YYPE

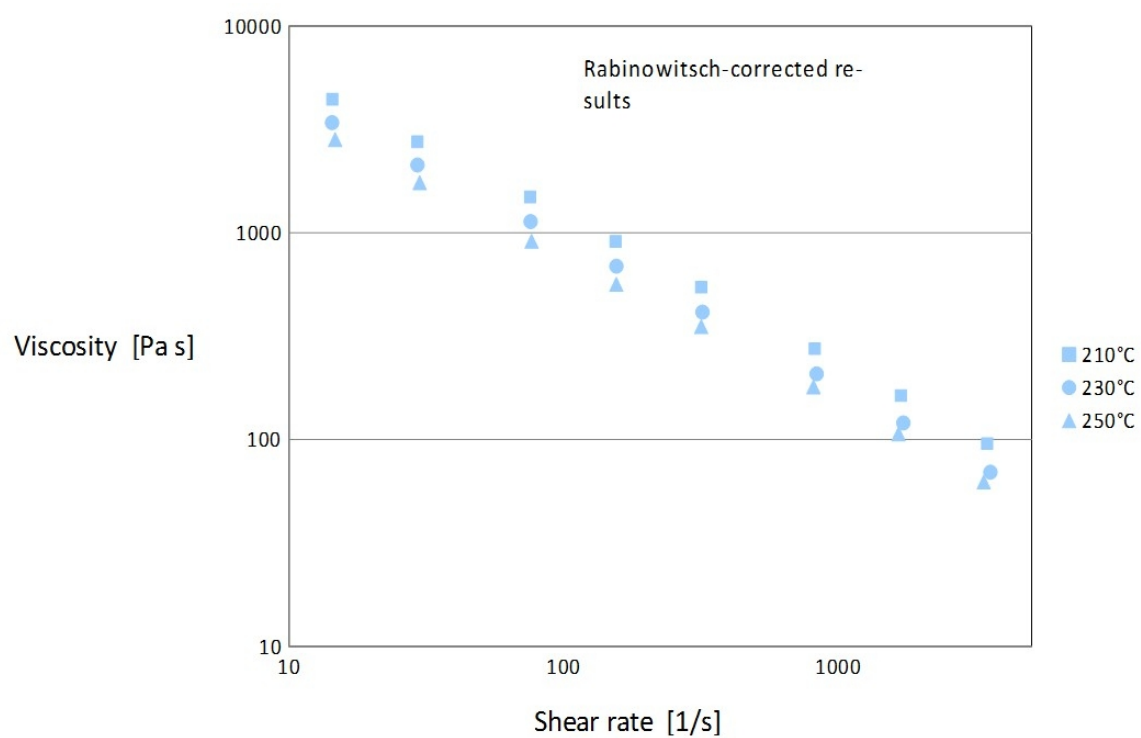


Obr. 26: Graf 3 – materiál FP201B

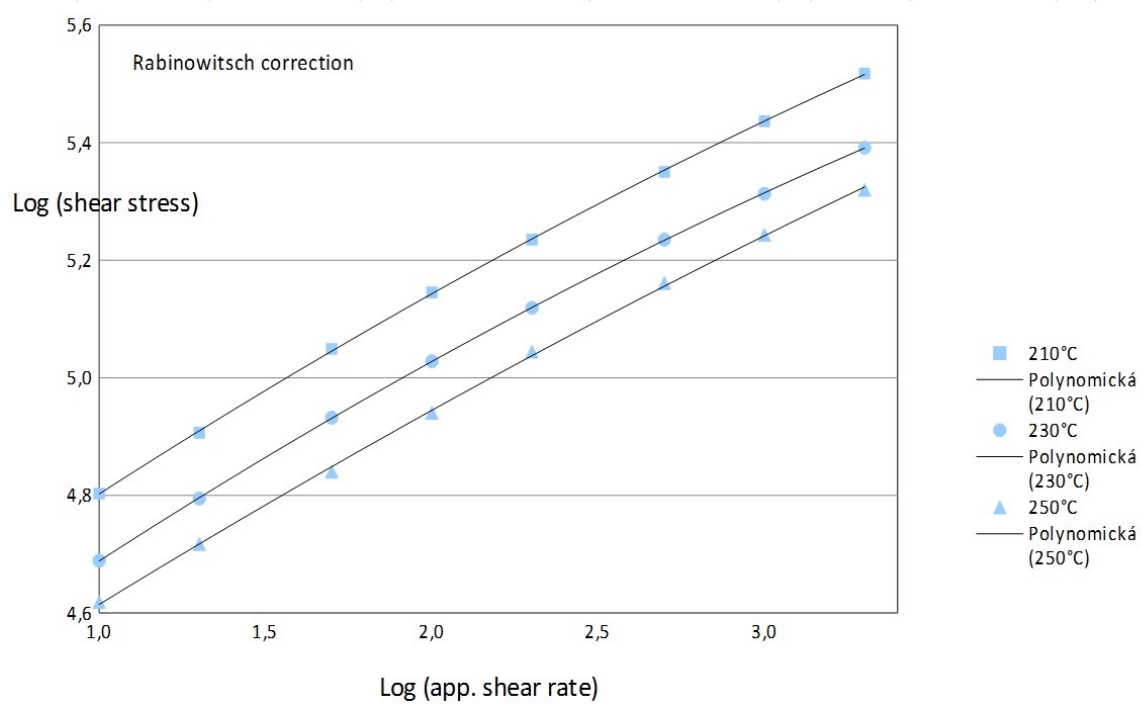
Viscosity (shear Rate) - Weissenberg/Rabinowitsch 30/2/180



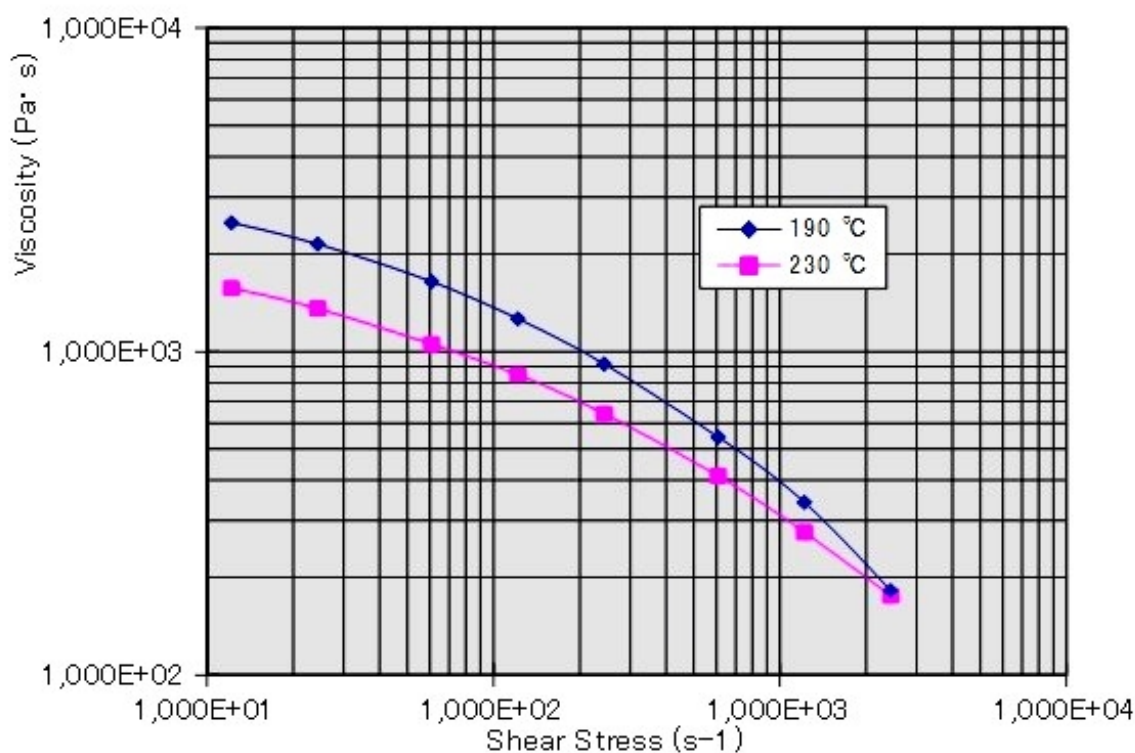
Obr. 27: Graf 4 – materiál ME3440



Obr. 28: Graf 5 – materiál TP12962



Obr. 29: Graf 6 – materiál TP12962



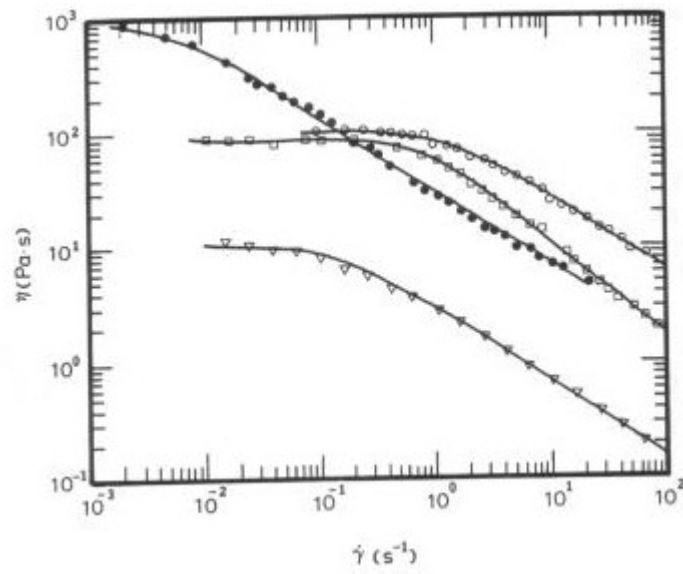
Obr. 30: Graf 7 – materiál NF498E

Na obrázku 24 je viskozitní graf materiálu Akulon® F136-C pro 3 různé teploty. Graf pochází z materiálového listu dodaného k projektu 5 vrstvé vytlačovací hlavy na trubky pro firmu Boco Pardubice Machines S.r.o.

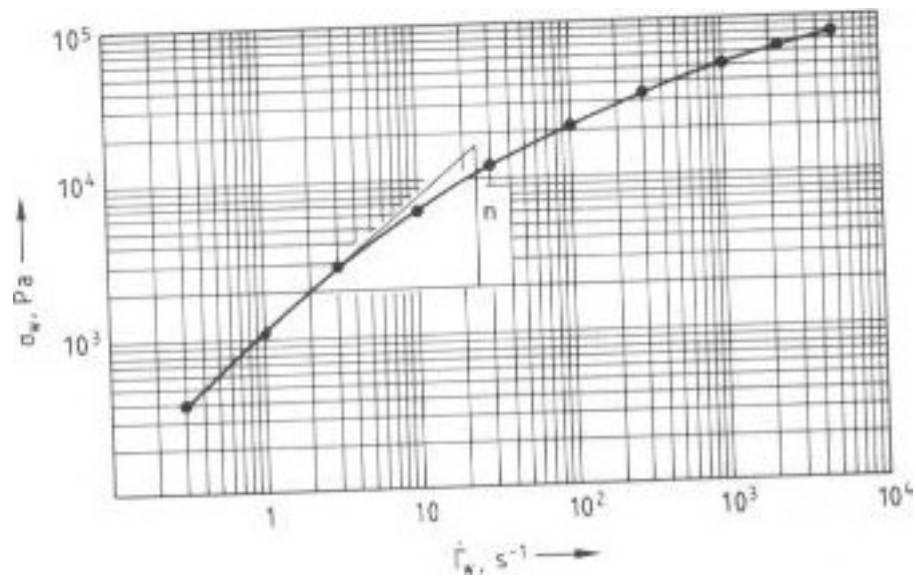
Graf na obrázku 25 pochází z projektu studie vytlačovacího stroje pro tureckou firmu Enformak Plastik Teknolojileri A.S.

Grafy na obrázcích 26, 27, 28, 29 a 30 pochází z projektu návrhu koextruzní hlavy na vyfukování fólie pro firmu KPSystem.

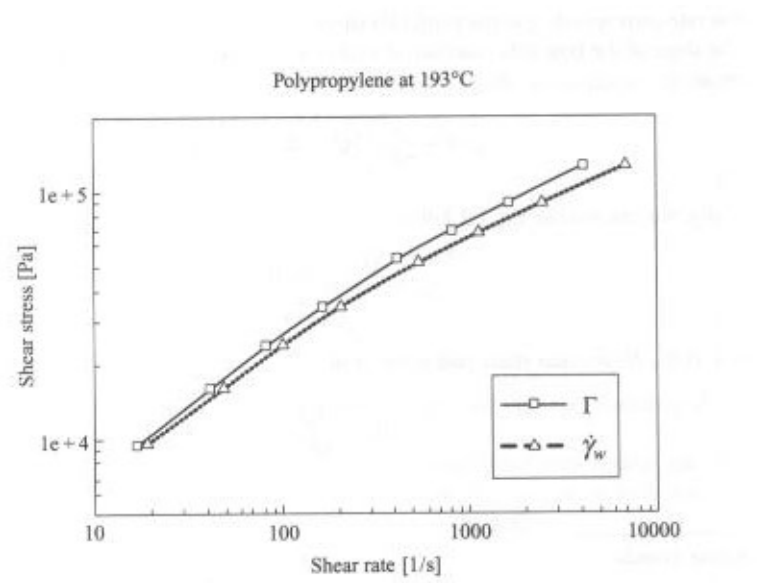
Ostatní grafy na obrázcích 31, 32 a 33 pochází z odborné literatury.



Obr. 31: Graf 8 [20]



Obr. 32: Graf 9 [21]



Obr. 33: Graf 10 [22]