

# ZÁVĚREČNÁ STUDIJNÍ PRÁCE

## dokumentace

### SportVision - Videoanalýza sprinterského výkonu



**Autor:** Filip Hřivňacký  
**Obor:** 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  
se zaměřením na počítačové sítě a programování  
**Třída:** IT4  
**Školní rok:** 2024/25



## **Poděkování**

Prostor k poděkování (například vedoucímu práce).

## **Prohlášení**

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým a prezentačním účelům na Střední průmyslové a umělecké škole v Opavě, Praskova 399/8.

V Opavě 1. 1. 2025

.....  
Podpis autora



# **Abstrakt**

Tato práce se zabývá vývojem softwarového nástroje SportVision, který slouží k analýze rychlosti a pohybu sportovce pomocí běžně dostupného videozáznamu. Cílem práce je vytvořit dostupnou alternativu k drahým hardwarem měřícím systémům, jako jsou fotobuňky. Aplikace využívá moderní metody počítacového vidění, konkrétně knihovny OpenCV a MediaPipe, pro detekci lidského těla (Pose Estimation) a následně vypočítává metriky, jako je průměrná rychlosť či čas na vymezeném úseku. Součástí práce je i implementace kalibrace scény a korekce perspektivy pro zvýšení přesnosti měření v reálných podmínkách.

## **Klíčová slova**

Počítacové vidění, Python, MediaPipe, OpenCV, analýza sportu, sprint, měření rychlosti, videoanalýza.

# **Abstract**

This thesis focuses on the development of the software tool SportVision, designed for analyzing athlete speed and movement using standard video footage. The goal is to create an accessible alternative to expensive hardware measurement systems. The application utilizes computer vision methods, specifically OpenCV and MediaPipe libraries, for human pose estimation and calculates metrics such as average speed or time over a specific distance. The thesis also addresses scene calibration and perspective correction to improve measurement accuracy in real-world athletic track conditions.

## **Keywords**

Computer Vision, Python, MediaPipe, OpenCV, sports analysis, sprint, speed measurement, video analysis.

# Obsah

<b>Úvod</b>	<b>3</b>
<b>1 Analýza problému a cíle práce</b>	<b>5</b>
1.1 Existující řešení pro sběr dat . . . . .	5
1.2 Přínos projektu SportVision . . . . .	6
<b>2 Teoretická východiska a použité technologie</b>	<b>7</b>
2.1 Princip počítačového vidění . . . . .	7
2.2 Konvoluční neuronové sítě (CNN) . . . . .	8
2.3 MediaPipe a model BlazePose . . . . .	9
2.4 Knihovna OpenCV . . . . .	9
<b>3 Tipy k psaní</b>	<b>11</b>
3.1 Základy . . . . .	11
3.2 Pokročilejší tipy . . . . .	15
<b>4 Když dokončují práci</b>	<b>17</b>
<b>A Spot diagramy a další</b>	<b>23</b>



# ÚVOD

Moderní sport v 21. století je neodmyslitelně spjat s daty a jejich analýzou. Schopnost přesně změřit výkon, analyzovat techniku a sledovat progres je tím, co často odlišuje amatérské snažení od profesionálního přístupu. V disciplínách, jako jsou sprinty, kde o vítězství rozhodují setiny sekundy, je přesné měření klíčové.

Profesionální sportovci a týmy mají k dispozici špičkové vybavení – od laserových bran přes motorizované odporové zařízení až po laboratoře biomechaniky. Tyto technologie jsou však pro běžné trenéry mládeže, amatérské atlety nebo školní týmy často finančně nedostupné. V praxi tak zůstávají odkázáni na ruční stopky, jejichž přesnost je silně ovlivněna reakční dobou měřícího, a na subjektivní vizuální hodnocení techniky bez možnosti přesné kvantifikace.

Tato práce si klade za cíl tento problém řešit. Představuje projekt *SportVision* – softwarový nástroj, který demokratizuje přístup ke sportovní analýze. S využitím běžně dostupného hardwaru (osobní počítač, kamera chytrého telefonu) a moderních metod počítačového vidění umožňuje přesně změřit rychlosť běžce a vizualizovat jeho pohyb bez nutnosti investic do drahého vybavení.

Práce je rozdělena do tří hlavních částí. První část se věnuje teoretickému rozboru problematiky a popisu použitých technologií, zejména knihoven OpenCV a MediaPipe, které stojí za detekcí pohybu. Druhá část detailně popisuje návrh a implementaci samotné aplikace v jazyce Python, včetně řešení technických výzev, jako je kalibrace scény a korekce perspektivy. Závěrečná část pak shrnuje výsledky testování aplikace v reálných podmírkách atletického stadionu a hodnotí její přesnost a použitelnost v praxi.

Výsledkem práce je funkční prototyp aplikace, který demonstruje, že pokročilá sportovní analýza nemusí být výsadou elitních týmů, ale může být dostupná komukoliv s notebookem a nadšením pro sport.



# 1 ANALÝZA PROBLÉMU A CÍLE PRÁCE

Cílem této práce je vytvořit dostupný nástroj pro analýzu sportovního výkonu, který eliminuje nutnost nákupu drahého hardwarového vybavení. Pro pochopení kontextu je nutné nejprve analyzovat dvě rozdílné technologie, které se v současné profesionální atletice používají.

## 1.1 EXISTUJÍCÍ ŘEŠENÍ PRO SBĚR DAT

V současné praxi se odděluje měření výkonu (rychlosť/čas) a analýza techniky (pohyb těla). Pro každou z těchto disciplín existují specializovaná zařízení.

### 1.1.1 Elektronická časomíra

Standardem pro přesné měření času a průměrné rychlosti jsou tzv. **fotobuňky (timing gates)**. Tyto systémy fungují na principu infračervených vysílačů a přijímačů, které jsou rozmístěny na trati ve specifických vzdálenostech (např. 10, 20, 30 metrů).

Časomíra se spustí a zastaví v momentě, kdy běžec fyzicky přeruší neviditelný paprsek mezi stojany. Výsledkem je přesný časový údaj, ze kterého lze vypočítat průměrnou rychlosť na daném úseku. Hlavní nevýhodou je, že systém nevidí běžce mezi bránami a neposkytuje žádnou informaci o jeho technice.

### 1.1.2 Optické systémy

Pro detailní analýzu biomechaniky (kostry a pohybu kloubů) se využívají systémy **Motion Capture**. V tomto případě musí mít sportovec na sobě speciální přiléhavý oblek s reflexními značkami (markery) umístěnými na klíčových kloubech.

Soustava infračervených kamer snímá polohu těchto značek a software následně rekonstruuje 3D model kostry sportovce. Ačkoliv jde o nejpřesnější metodu analýzy techniky, její nasazení v běžném tréninku je téměř nemožné kvůli extrémní ceně, nutnosti laboratorních podmínek a zdlouhavé přípravě sportovce (oblékání do senzorů).

## 1.2 PŘÍNOS PROJEKTU SPORTVISION

Aplikace SportVision byla navržena tak, aby funkce obou výše zmíněných systémů integrovala do jednoho softwarového řešení. Její hlavní přínos spočívá ve třech oblastech:

1. **Spojení času a obrazu:** Projekt nahrazuje fotobuňky virtuální detekcí průchodu a zároveň nahrazuje drahý Motion Capture využitím umělé inteligence, která detektuje kostru bez nutnosti speciálního obleku .
2. **Demokratizace sportovní analýzy:** Nahrazením fyzických senzorů a kamerových systémů běžným notebookem se technologie stává dostupnou pro širokou veřejnost – od učitelů tělocviku po trenéry mládeže.
3. **Flexibilita:** Systém nevyžaduje fyzické rozestavování bran na dráze ani oblékání sportovce do senzorů. Měření lze provést okamžitě a neinvazivně.

## **2 TEORETICKÁ VÝCHODISKA A POUŽITÉ TECHNOLOGIE**

Zpracování obrazu a detekce pohybu jsou disciplíny spadající pod obor počítačového vidění (Computer Vision). Tato kapitola popisuje principy neuronových sítí a konkrétní softwarové nástroje, které byly v projektu využity.

### **2.1 PRINCIP POČÍTAČOVÉHO VIDĚNÍ**

Počítačové vidění je oblast informatiky, která se snaží naučit počítače „vidět“ a interpretovat vizuální data podobným způsobem jako člověk. Pro počítač je digitální obraz reprezentován jako matice čísel (pixelů).

U barevného obrazu se nejčastěji využívá model RGB, kde je každý pixel definován třemi hodnotami (červená, zelená, modrá) v rozsahu 0–255. Úkolem algoritmů je v této záplavě čísel nalézt vzory, hrany a objekty. V projektu SportVision je tento princip aplikován na video, což je ve své podstatě sekvence statických obrázků (snímků) přehrávaných vysokou rychlostí.

## 2.2 KONVOLUČNÍ NEURONOVÉ SÍTĚ (CNN)

Zatímco klasické algoritmy počítačového vidění spoléhaly na ručně definovaná pravidla (např. „hledej světlý kruh jako hlavu“), moderní detekce, kterou využívá i tento projekt, je založena na hlubokém učení (Deep Learning) a **konvolučních neuronových sítích (CNN)**.

CNN jsou inspirovány biologickou strukturou zrakové kůry v mozku. Sítě se skládají z mnoha vrstev, kterými obraz prochází:

1. **Konvoluční vrstvy:** Aplikují na obraz filtry, které detekují jednoduché prvky, jako jsou hrany nebo barevné přechody.
2. **Pooling vrstvy:** Snižují rozlišení dat a zachovávají jen nejdůležitější informace, čímž snižují výpočetní náročnost.
3. **Plně propojené vrstvy:** V hlubších částech sítě se již nedetekují jen hrany, ale komplexní tvary (oči, končetiny), které sítě na základě tréninku klasifikují jako „člověka“.

## 2.3 MEDIAPIPE A MODEL BLAZEPOSE

Jádrem detekce pohybu v této práci je framework **MediaPipe** od společnosti Google, konkrétně jeho řešení pro detekci postoje zvané **BlazePose**. Tento model byl vybrán pro svou unikátní architekturu, která umožňuje běh v reálném čase i na běžných procesorech (CPU) bez nutnosti dedikované grafické karty.

### 2.3.1 Architektura detektor-tracker

Většina starších systémů analyzuje každý snímek videa od nuly, což je výpočetně náročné. MediaPipe využívá chytřejší přístup sestávající ze dvou kroků:

1. **Detektor (Pose Detector):** Tato část sítě běží pouze na začátku nebo když se ztratí sledování. Její úkol je najít v obrazu oblast zájmu (ROI – Region of Interest), kde se nachází člověk. Protože je detekce celého těla náročná, BlazePose nejprve hledá obličeji (který má specifické rysy) a od něj odvodí polohu zbytku těla.
2. **Sledovač (Pose Landmarker):** V dalších snímcích už se nepoužívá detektor. Sítě pouze předvírá, kam se klíčové body posunuly z předchozí pozice. Tím se ušetří obrovské množství výkonu. Pokud si je model jistý, že body stále sleduje, detektor se nespouští.

### 2.3.2 Topologie landmarků

Model BlazePose detekuje celkem **33 klíčových bodů (landmarks)** ve 3D prostoru. Každý bod má souřadnice  $x, y$  (poloha v obrazu) a  $z$  (hloubka vůči středu boků).

Pro účely měření sprintu v aplikaci SportVision jsou klíčové tyto body:

- **Landmarky 11 a 12 (Ramena):** Jejich průměr určuje střed hrudníku, což je v atletice rozhodující bod pro protnutí cílové čáry.
- **Landmarky 23 a 24 (Kyče):** Definují střed těla a těžiště.
- **Landmarky 27–32 (Kotníky a chodidla):** Ačkoliv nejsou využity pro měření času, jsou vykreslovány pro vizuální kontrolu techniky běhu.

## 2.4 KNIHOVNA OPENCV

OpenCV (Open Source Computer Vision Library) je podpůrnou, ale nezbytnou technologií projektu. Zatímco MediaPipe provádí „inteligentní“ analýzu, OpenCV obstarává „hrubou práci“ s obrazovými daty:

- **I/O operace:** Načítání videosouborů (.mp4) a extrakce jednotlivých snímků.
- **Preprocessing:** Konverze barevného prostoru z BGR (standard OpenCV) na RGB (vyžadováno MediaPipe).
- **Vizualizace:** Vykreslování vektorové grafiky (čáry, kružnice, text) přímo do obrazových dat na základě souřadnic vypočítaných modelem MediaPipe.



## 3 TIPY K PSANÍ

Jak už jsem psal výše,  $\text{\LaTeX}$  je dosti komplexní systém, který umožňuje psát velmi rozsáhlé texty. Jeho autor Donald Knuth ho stvořil, aby mohl vydat svou učebnici *The Art of Computer Programming* a dodnes je využíván pro sazbu skript, učebnic, článků či závěrečných prací. V této kapitole najdeš ukázky různých funkcí a balíčků  $\text{\LaTeX}u$  od těch nejzákladnějších až po složitější. Neznamená to nutně, že všechny musíš použít, ale když potřebuješ pomoci, tak je dobré mít oporu. Pokud s  $\text{\LaTeX}em$  úplně začínáš, tak ti můžu doporučit příručku *Ne příliš stručný úvod do systému \LaTeX{}2e* [2]. Případně spoustu užitečných informací nalezneš na Wikibooks [3]. Pokud narazíš na nějaký problém, googli. Na internetu je spousta fór, kde pravděpodobně už někdo podobný problém řešil. Asi nejvíce o tom najdeš na stránce *TeX - LaTeX Stackexchange* [4].

### 3.1 ZÁKLADY: TEXT, OBRÁZKY, TABULKY A CITACE

Psaní v  $\text{\LaTeX}u$  není žádná věda, stačí psát normálně do zdrojového souboru. Pokud bys chtěl psát odrážky či číslovaný seznam, pak můžeš použít prostředí `itemize` či `enumerate`. Často je důležité používat nezlohitelnou mezeru. Tu uděláš pomocí `~` (tildy). Pokud budeš chtít psát uvozovky použij příkaz `uv`, pomocí něj se ti vytvoří uvozovky podle příslušného jazyka. V česku tedy ve formátu 99 66. Použití příkazu najdeš níže v textu. Občas je zapotřebí  $\text{\LaTeX}u$  pomoci při rozdělování slov. To se udělá snadno vložením symbolů `\-` mezi jednotlivé slabiky.

#### 3.1.1 Tabulky

U tabulek platí to stejné co u obrázků. Zarovnávají se na střed a nechávají se „plavat“ v textu. Tabulka narozdíl od textu, má popisek nahoře. U tabulky 3.1 je použit balíček `booktabs`, pomocí kterého je celá tabulka naformátovaná. Seznam jak obrázků tak tabulek je pak vytvořen pomocí příkazů `listoftables` a `listoffigures` na konci práce před literaturou.

#### 3.1.2 Obrázky

Tabulka 3.1: Tato tabulka slouží jako ukázka toho, jak mohou tabulky vypadat.

záhlaví	této	tabulky
obsah	tabulky	už
není	oddělený	čarami

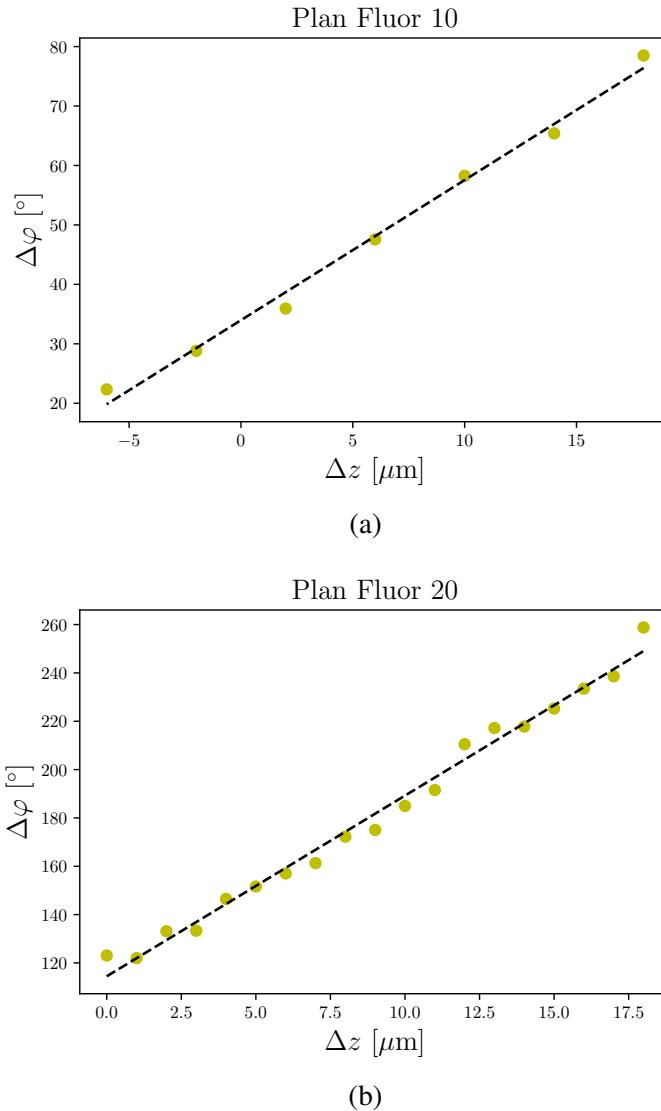
U obrázků je dobré používat vektorové formáty, pokud to jde. L<sup>A</sup>T<sub>E</sub>X se nejvíc kamarádí s formátem PDF. Do známého PDFka lze z jiných vektorových formátů (ať už SVG či ESP) obrázky přenést snadno pomocí grafických programů, jako je třeba Inkscape. L<sup>A</sup>T<sub>E</sub>X si rozhodně poradí i s tradičními formáty PNG a JPG, avšak tyto obrázky mohou zabírat více prostoru a při tisku se může projevit nižší rozlišení obrázků. Pokud chceš používat tyto obrázky, rozhodně měj na paměti, aby měly rozlišení alespoň 250, ideálně 330 ppi. Obrázky se vkládají do prostředí `figure`, při úpravě šířky je možné krom tradičních jednotek jako cm nebo mm použít také jako jednotku šířku stránky `textwidth` to se hodí zejména když chceš mít více podobrázků. U každého obrázku je důležité aby měl popisek, `caption`. Do popisku napiš, co na obrázku je, případně nějaký další popis, tak aby čtenář následně neměl sebemenší pochybnost. U obrázků, co nejsou tvoje, nezapomeň na citaci. Jinak by to totiž znamenalo, že jsi obrázek dělal ty sám, což není etické přivlastňovat si cizí díla. Popisek obrázku je věta, proto musí vždy končit tečkou.



Obrázek 3.1: Logo SŠPU Opava [5].

Když chceš odkazovat na obrázek, stačí pak už jen napsat příkaz `ref` a do závorek napsat označení obrázku. Třeba logo SOČky, můžeš vidět na obrázku 3.1 [?].

Pokud bys měl více podobrázků přichází do hry balíček `subcaption`. Pomocí něj lze vysázenet i podobrázky. U podobrázků se popisek píše pouze jeden, dolů. Je v tomto případě vhodné použít navíc hranaté závorky, do nichž se napíše kratší popisek, který se následně ukáže v seznamu obrázků. Všimni si, že obrázky jsou naschvál široké. Je to proto, aby byly dobře čitelné. Také si všimni popisku grafů. Ačkoli nejspíš netušíš co je to DH PSF či defokusace objektivu mělo by ti být jasné, že je důležité přesně graf popsat. To znamená co je na vodorovné ose, co je na svislé ose. V jakých jednotkách veličiny jsou. Které body co znamenají, která křivka má jaký význam. Napsat samotné „ $\Delta\phi$ “ je málo, vždy raději připoměň, co daná značka znamená.



Obrázek 3.2: Graf závislosti rotace DH PSF  $\Delta\varphi$  na defokusaci objektivu  $\Delta z$ , (a) při použití objektivu Plan Fluor 10, (b) při použití objektivu Plan Fluor 20. Měřená data (žluté body) jsou lineárně proloženy (přerušovaná přímka).

### 3.1.3 Literatura

V L<sup>A</sup>T<sub>E</sub>Xu lze dělat seznam literatury dvěma způsoby. V této šabloně jsem použil ten, kdy se seznam literatury píše přímo do práce. Pro jeho vygenerování doporučuji použít některý z generátorů, jako jsou například Citace PRO [6]. Pomocí citací lze vygenerovat přímo dokument, který se pak už jen překopíruje do textu a člověk nemusí nic zvýrazňovat. Dále lze využít Bibtex, který rozhodně do budoucna hodlám zaimplementovat do šablony, avšak jeho použití nemusí být tak přátelské k začátečníkům. Pokud bys chtěl odkazovat na vícero zdrojů stačí je napsat vedle sebe oddělené čárkou [2, 6, 7]. Případně můžu odkaz na konkrétní stránku dát do hranatých závorek, viz [7, str. 1]

### 3.1.4 Programový kód

Pro vložení programového kódu do dokumentu LaTeX s možností zvýraznění syntaxe můžete použít balíček `listings`. Tento balíček nabízí široké možnosti pro formátování kódu, včetně zvýraznění syntaxe pro různé programovací jazyky. Nejprve je třeba do preambule LaTeX dokumentu přidat

`\usepackage{listings}` a nastavit příslušné parametry. Příklad nastavení pro jazyk Python by mohl vypadat takto:

```
1 # Python code here
2 def hello_world():
3     print("Hello, world!")
4
```

Kód 3.1: Ukázka Python kódu

```
1 // JavaScript code here
2 function helloWorld() {
3     console.log("Hello, world!");
4 }
5
```

Kód 3.2: Ukázka JS kódu

```
1 /* eslint-env es6 */
2 /* eslint-disable no-unused-vars */
3
4 import Axios from 'axios'
5 import { BASE_URL } from './utils/api'
6 import { getAPIToken } from './utils/helpers'
7
8 export default class User {
9     constructor () {
10         this.id = null
11         this.username = null
12         this.email = ''
13         this.isActive = false
14         this.lastLogin = '' // ISO 8601 formatted timestamp.
15         this.lastPWChange = '' // ISO 8601 formatted timestamp.
16     }
17 }
```

```

18
19   const getUserProfile = async (id) => {
20     let user = new User()
21     await Axios.get(
22       `${BASE_URL}/users/${id}`,
23     {
24       headers: {
25         'Authorization': `Token ${getAPIToken()}`,
26       }
27     }
28   ).then(response => {
29     // ...
30   }).catch(error => {
31     // ...
32   })
33 }
34

```

Kód 3.3: ES6 (ECMAScript-2015) Listing

## 3.2 POKROČILEJŠÍ TIPY, KTERÉ SE MOHOU HODIT

### 3.2.1 Rovnice

Sazba matematiky je věda sama o sobě. Ačkoli Word prošel obrovskou změnou a je v tomto mnohem lepší, tak L<sup>A</sup>T<sub>E</sub>X je pro to přímo (ještě jsem neviděl matematika, co by používal Word). Spolu s balíčky amsmath a amsfonts snad neexistuje nic, co by se používalo a L<sup>A</sup>T<sub>E</sub>X by to nezvládl. Ať už jde o základní věci jako řecká písmenka –  $\alpha, \beta, \gamma, \dots$  – integrály –  $\int_{l_i}^{l_f} \tau dl$  – až třeba po speciální písmena –  $\mathcal{F} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ . Pro případ, že bys potřeboval nějaké speciální integrály, je tu balíček esint, pomocí něj můžeš napsat třeba

$$\iint_{S(V)} \vec{E} \cdot d\vec{S} = \iiint_V (\vec{\nabla} \cdot \vec{E}) dV.$$

Jak můžeš vidět tak rovnice lze psát jednak do textu a nebo pokud se jedná o nějakou důležitou nebo rozsáhlější rovnici tak na samostatný řádek. Pokud je rovnice opravdu důležitá, tak je vhodné ji také číslovat. Pak se na ni můžeš dále odkazovat v textu.

$$\vec{F} = m\vec{a} \tag{3.1}$$

... Například podle druhého Newtonova zákona, rovnice (3.1) ... Zároveň je vždy nutné vysvětlit co která veličina znamená. V tomto případě bych napsal, že v druhém Newtonově zákoně vektor síly  $\vec{F}$  odpovídá součinu hmotnosti tělesa  $m$  a jeho zrychlení  $\vec{a}$ . Věřím, že se sazbou matematiky ti pomůže tvůj školitel, případně mi můžeš napsat (mail je v úvodu). Jednotlivé funkcionality spolu se seznamem znaků nalezneš jednak v Ne příliš stručném úvodu [2] nebo na Wikibooks v sekcích *Mathematics* a *Advanced mathematics* [3].

## **4 KDYŽ DOKONČUJI PRÁCI**

Každou práci je dobré zkontořovat, aby v ní nebyly pravopisné chyby, nebyla těžkopádně napsaná – byla čtivá – a neobsahovala žádný typografický nedostatek. Proto, když práci sepíšeš, nech ji chvilku odležet, třeba týden. Pak si ji po sobě znova přečti. Hned uvidíš, kolik věcí bys napsal jinak případně kde tě bije do očí jaká chyba. Dej práci přečíst také svému školiteli a případně češtinaři. Zajistíš tak, že bude obsahovat méně chyb. Pak můžeš práci vytisknout a hurá do soutěže.



## ZÁVĚR

Věřím, že jsem ti spolu se šablonou poskytl několik tipů, jak napsat práci. At' už jde o úplné začátky s L<sup>A</sup>T<sub>E</sub>Xem. Či ukázku toho, co vše s ním zvládneš. Pokud bys měl k šabloně libovolné dotazy, rozhodně se na mě obrat'. L<sup>A</sup>T<sub>E</sub>Xtvé práci dodá určitou krásu, tak doufám, že ti dodá sebevědomí a uspěješ při soutěži. A i kdyby ne vzpomeň si, kolik ses toho musel naučit a hned uvidíš o jaký kus ses posunul.



# LITERATURA

- [1] DOKULIL Jakub. *Šablona pro psaní SOČ v programu L<sup>A</sup>T<sub>E</sub>X* [Online]. Brno, 2020 [cit. 2020-08-24]. Dostupné z: [https://github.com/Kubiczek36/SOC\\_sablon](https://github.com/Kubiczek36/SOC_sablon)
- [2] OETIKER, Tobias, Hubert PARTL, Irene HYNA, Elisabeth SCHEGL, Michal KOČER a Pavel SÝKORA. *Ne příliš stručný úvod do systému LaTe<sub>X</sub>2e* [online]. 1998 [cit. 2020-08-24]. Dostupné z: <https://www.jaroska.cz/elearning/informatika/typografie/lshort2e-cz.pdf>
- [3] *Wikibooks: L<sup>A</sup>T<sub>E</sub>X* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2020-08-24]. Dostupné z: <https://en.wikibooks.org/wiki/LaTeX>
- [4] *TeX - L<sup>A</sup>T<sub>E</sub>X Stack Exchange* [online]. Stack Exchange, 2020 [cit. 2020-09-01]. Dostupné z: <https://tex.stackexchange.com>
- [5] *Střední škola průmyslová a umělecká Opava* [online]. [cit. 2023-11-11]. Dostupné z: <https://www.sspu-opava.cz>
- [6] *Citace PRO* [online]. Citace.com, 2020 [cit. 2020-08-31]. Dostupné z: <https://www.citacepro.com>
- [7] BORN, Max a Emil WOLF. *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light*. 7th (expanded) edition. Reprinted with corrections 2002. 15th printing 2019. Cambridge: Cambridge University Press, 2019. ISBN 978-0-521-64222-4.

# Seznam obrázků

3.1 Logo SŠPU Opava [5]. . . . .	12
3.2 Graf závislosti rotace DH PSF $\Delta\phi$ na defokusaci objektivu $\Delta z$ . . . . .	13

# Seznam tabulek

3.1 Tato tabulka slouží jako ukázka toho, jak mohou tabulky vypadat. . . . .	12
--	----

## **PŘÍLOHA A SPOT DIAGRAMY A DALŠÍ**