



ZADÁNÍ BAKALÁ SKÉ PRÁCE

| | |
|--------------------------|--------------------------------------|
| Název: | Analýza a popis vzorku pneumatiky |
| Student: | Radka Vopatová |
| Vedoucí: | doc. RNDr. Ing. Marcel Ji ina, Ph.D. |
| Studijní program: | Informatika |
| Studijní obor: | Teoretická informatika |
| Katedra: | Katedra teoretické informatiky |
| Platnost zadání: | Do konce zimního semestru 2018/19 |

Pokyny pro vypracování

Cílem práce je analyzovat r zné typy vzork pneumatik a jejich otisk a následn navrhnout zp sob jejich popisu (reprezentace) tak, aby se následn daly srovnávat r zné vzorky a otisky mezi sebou a dále navrhnout vhodnou metriku pro srovnávání t chto vzork a otisk , tj. stanovování míry jejich podobnosti. Sou ástí práce je i sb r snímk r zných pneumatik a jejich otisk a jejich následné zpracování, tj. extrakce pot ebných charakteristik pro jejich popis a srovnávání.

- 1) Prove te rešerši stávajících metod popisu vzorku pneumatiky. Inspirujte se popisem otisku prstu.
- 2) Navrhn te zp sob reprezentace vzorku, resp. otisku pneumatiky a zp sob porovnávání vzork (otisk) mezi sebou.
- 3) Navržené zp soby implementujte ve vhodném programovacím jazyce. Umožn te ru ní, p ípadn poloautomatický p evod obrazových dat do navržené reprezentace.
- 4) Prove te sb r obrazových dat vzork a otisk pneumatik.
- 5) Navržené a implementované postupy ov te na reálných datech a dosažené výsledky vyhodno te.

Seznam odborné literatury

Dodá vedoucí práce.

doc. Ing. Jan Janoušek, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
d kan

V Praze dne 28. února 2017

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



Bakalářská práce

Analýza a popis vzorku pneumatiky

Radka Vopatová

Vedoucí práce: doc. RNDr. Ing. Marcel Jiřina, Ph.D.

8. ledna 2018

Poděkování

Děkuji doc. RNDr. Ing. Marcelu Jiřinovi, Ph.D. za vedení této práce, za jeho cenné rady a vstřícný přístup. Také děkuji rodině, přátelům a všem, kteří mě podpořili při tvorbě této práce i v samotném studiu.

Prohlášení

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

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Praze dne 8. ledna 2018

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2018 Radka Vopatová. Všechna práva vyhrazena.

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

Odkaz na tuto práci

Vopatová, Radka. *Analýza a popis vzorku pneumatiky*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018.

Abstrakt

Tato bakalářská práce se zabývá analýzou a popisem vzorku pneumatiky z fotografie otisku. Cílem práce je seznámit se s metodami popisu otisků používaných v současné době a návrh vlastního způsobu reprezentace vzorku. Reprezentace bude implementovaná v prostředí MATLAB a následně testována na reálných datech.

Klíčová slova analýza, pneumatika, dezén, běhoun, otisk pneumatiky, MATLAB

Abstract

This bachelor thesis deals with the analysis and description of tire tread imprint from photograph. The goal is to explore recent methods which are used for description of tire tread pattern and design own method for its representation. Selected representation will be implemented in MATLAB environment and then it will be tested on real data.

Keywords analysis, tire, tire tread, tread pattern, tread imprint, MATLAB

Obsah

| | |
|-----------------------------------|-----------|
| Úvod | 1 |
| 1 Představení úlohy | 3 |
| 1.1 Pneumatika | 3 |
| 1.2 Běhoun | 5 |
| 1.3 Forezní využití | 7 |
| 2 Analýza a návrh | 11 |
| 2.1 Nástrahy zpracování | 11 |
| 2.2 Algoritmy | 11 |
| 2.3 Existující rozhraní | 17 |
| 2.4 Návrh | 23 |
| 3 Realizace | 31 |
| 3.1 Databáze | 31 |
| 3.2 Implementace | 33 |
| 4 Testování a vyhodnocení | 37 |
| 4.1 Testování | 37 |
| 4.2 Vyhodnocení | 38 |
| Závěr | 39 |
| Literatura | 41 |
| A Seznam použitých zkratk | 45 |
| B Uživatelská příručka | 47 |
| C Obsah přiloženého CD | 49 |

Seznam obrázků

| | | |
|------|--|----|
| 1.1 | Konstrukce pneumatiky[1] | 4 |
| 1.2 | Druhy pneumatik[2] | 5 |
| 1.3 | Názvosloví | 5 |
| 1.4 | Typy dezénu (zleva symetrický, asymetrický a směrový)[3] | 7 |
| 1.5 | Příklad sekvence úseků pro snížení hlučnosti[4] | 8 |
| | | |
| 2.1 | Struktura 1D DT-CWT filtru[5] | 13 |
| 2.2 | Geometrické závislosti 2D Radonové transformace[6] | 14 |
| 2.3 | Vývojový diagram algoritmu Radon-DTCWT[5] | 15 |
| 2.4 | Frekvenční mapa Curvelet transformace[5] | 16 |
| 2.5 | Druhy vzorků pneumatiky (zleva vertikální, horizontální, diagonální a nepravidelný)[5] | 16 |
| 2.6 | Vývojový diagram algoritmu CEDA[5] | 17 |
| 2.7 | Databáze v EXCEL formátu[7] | 18 |
| 2.8 | Databáze v ACCESS formátu[7] | 18 |
| 2.9 | Ukázka GUI TreadMate[8] | 20 |
| 2.10 | Ukázka nalezených pneumatik[8] | 20 |
| 2.11 | Struktura a obsah databáze[9] | 22 |
| 2.12 | Rozhraní databáze[9] | 22 |
| 2.13 | Konceptuální model databáze | 24 |
| 2.14 | Návrh GUI | 26 |
| 2.15 | Předzpracování: (a) originál, (b) po rotaci, (c) po vyříznutí | 27 |
| 2.16 | Algoritmus pro celý otisk | 28 |
| 2.17 | Volba shody | 28 |
| | | |
| 3.1 | Označené charakteristiky ve vyvíjené aplikaci | 31 |
| 3.2 | Ukázka záznamu z databáze | 33 |
| 3.3 | Ukázka práce s GUIDE | 33 |
| 3.4 | Grafické rozhraní Application Compiler | 35 |
| 3.5 | Výsledný produkt | 36 |

| | | |
|-----|--|----|
| B.1 | Příklad vstupu (zleva špatně, správně) | 47 |
| B.2 | Příklad vstupu (zleva špatně, správně) | 47 |

Seznam tabulek

| | | |
|-----|--|----|
| 2.1 | Popis jednotlivých pozic vektoru | 25 |
|-----|--|----|

Úvod

V dnešní době jsou automobily všudypřítomné, téměř každá rodina alespoň jeden vlastní. Jen od roku 2006 do roku 2015 narostl počet celosvětově užívaných vozidel téměř o 30%. [10]

I když automobily značně ulehčují život v moderní společnosti, přináší to i svou daň. S nárůstem užívání vozidel roste i počet nehod zapříčiněných právě těmito výdobytky moderní doby. Mnohem více jsou vozidla zapojena i do kriminální činnosti, a proto vznikají databáze vozidel, které mají pomoci ve vyšetřování.

Identifikace otisků pneumatik z místa činu se v posledních 20 letech stalo velmi populární. [7] Identifikování specifického otisku je ovšem velice náročná disciplína. Vzhledem k množství druhů pneumatik na celosvětovém trhu, vyhledávání v tištěné databázi trvá dlouhou dobu. Ale díky pokroku IT technologií a elektronických zařízení, lze tento proces výrazně zefektivnit.

Tato práce se zabývá tematikou identifikace otisku pneumatiky. Snaží se co nejrychleji vyhledávat shodu v databázi s co nejvyšší přesností. V práci nechybí ani shrnutí současných způsobů řešení problémů a navržení vlastního postupu. Výstupem práce je funkční prototyp programu, který hledá shodu otisku ve vytvořené databázi.

Představení úlohy

Tato kapitola slouží k seznámení čtenáře se základními skutečnostmi, které budou tvořit stavební kameny pro tuto práci. Začíná u základních prvků z kterých je pneumatika tvořena a následně se blíže zaměřuje na běhoun pneumatiky. Je zde uvedeno základní rozdělení běhounu podle vzorků, a také je zde nastíněn význam jednotlivých prvků obsažených ve vzorku.

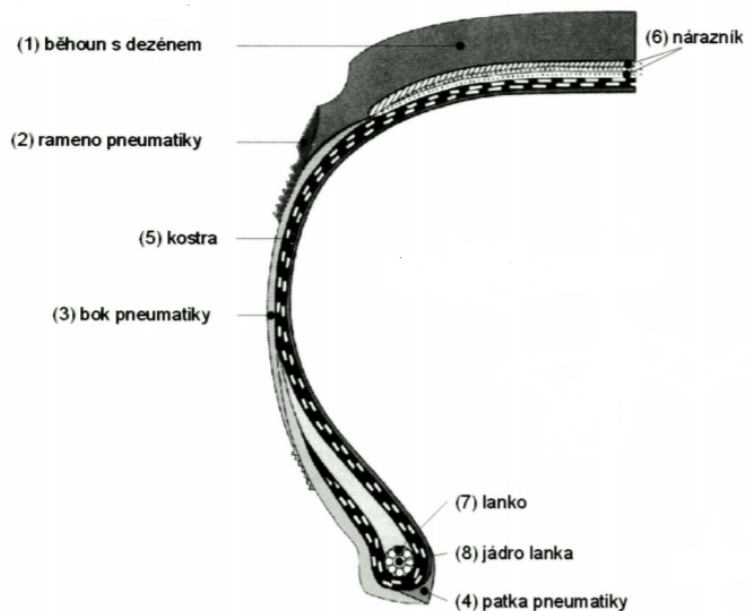
V poslední části této kapitoly je ukázáno k čemu slouží otisk běhounu z forenzního hlediska. A to sice podle jakých znaků dokáží kriminalisté přiřadit zkoumaný otisk ke konkrétní pneumatice.

1.1 Pneumatika

Pneumatikou je označován celek, který se skládá z pláště, duše a ochranné vložky. Konstrukce pneumatiky dělí její části na *8 prvků*:

1. běhoun s dezénem
2. rameno pneumatiky
3. bok pneumatiky
4. patka pneumatiky
5. kostra
6. nárazník
7. lanko
8. jádro lanka

Tato práce se bude především zabývat *běhounem* a to právě kvůli *dezénu* a jeho vzorům. Nicméně pro dokreslení obrazu celého problému budou okrajově zmíněny i další důležité prvky.



Obrázek 1.1: Konstrukce pneumatiky[1]

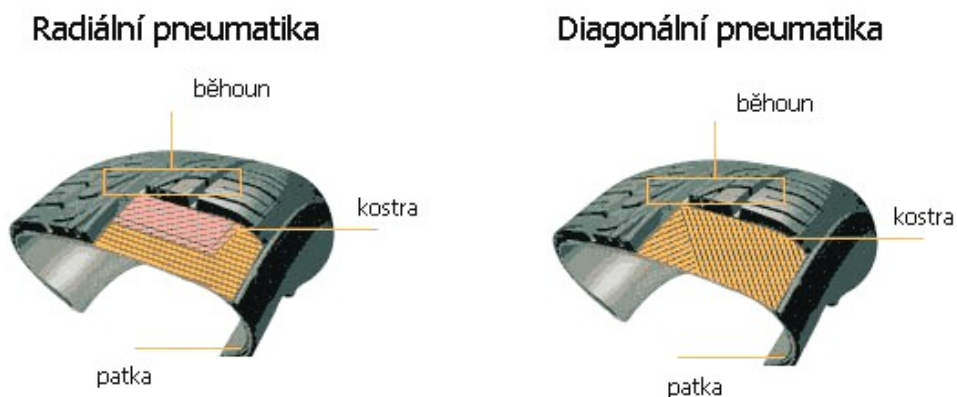
1.1.1 Druhy pneumatik

Kostra pneumatiky je tvořena z vložek bavlněných, plastových, polyesterových nebo ocelových vláken navzájem spojených pryží. Podle způsobu vedení těchto pásů rozeznáváme 3 *druhy* pneumatik:

- **Radiální** – v kostře pneumatiky jsou jednotlivé kordy poskládány rovnoběžně od patky k patce a svírají s obvodovou kružnicí úhel 90 stupňů, kostra je vyztužená nárazníkovými pásy
- **Diagonální** – v kostře pneumatiky se kordy v jednotlivých vrstvách vzájemně kříží a svírají s obvodovou kružnicí úhel 30 až 40 stupňů
- **Smíšené** – kombinace dvou předchozích

V dnešní době je většina pneumatik u osobních vozidel *radiálních*, protože jsou účinnější v redukci kroucení¹ povrchu v době používání (oproti smíšeným). Ale i tak stále můžeme *smíšené* pneumatiky najít např. u nákladních vozidel.

¹smršťování a roztahování



Obrázek 1.2: Druhy pneumatik[2]

1.2 Běhoun

Běhounem se rozumí ta část pláště, která je opatřena vzorkem a zajišťuje kontakt kola s vozovkou. Tento vzorek nazýváme **dezénem** a slouží k přenášení hnací, brzdné a boční síly. Dezén pneumatiky je vzájemným uspořádáním *výstupků, žebor, rýh, lamel, podélných a příčných drážek*.

Každý prvek designu má svůj účel, který má zásadní vliv na správnou činnost pneumatiky. V literatuře se názvosloví prvků značně liší, proto zde autorka vymezí pojmy, které budou používány po zbytek práce. [11]



Obrázek 1.3: Názvosloví

Obvodové drážky jsou rýhy vedoucí po celém obvodu pneumatiky (ve směru otáčení kola). Zajišťují přenos příčných sil, čímž ovlivňují směrovou stabilitu vozidla při jízdě a hrají významnou roli při odvodu vody ze styčné plochy pneumatiky s vozovkou.

Příčnými drážkami také nazýváme rýhy, ale v tomto případě se jedná o rýhy, které nevedou po celém obvodu pneumatiky. Zajišťují přenos tažné síly na vozovku a přenos sil vznikajících při brzdění. Efektivně odvádějí vodu při jízdě na mokré vozovce a starají se o co nejlepší záběr ve sněhu.

Blok je menší část dezénu ohraničená obvodovými a příčnými drážkami.

Lamelami se rozumí jemné rýhování jednotlivých částí na pneumatice, které zvyšuje záběrové a brzdící schopnosti. Při akceleraci či brzdění se rozevřou, čímž vznikne větší množství záběrových hran. Hlavní význam při jízdě na mokré vozovce spočívá v tom, že rozevřením se do vzniklých dutin může dostat voda, a tím se zmenší množství vody, které je nutno odvést ze styčné plochy pneumatiky s vozovkou. Při přenosu bočních sil se lamely zaklesnou a vytvoří tak blok s vyšší tuhostí, který je příčinou lepší boční stability a nižšího opotřebení.

Žebro je pás táhnoucí se po celém obvodu pneumatiky, často vymezený obvodovými drážkami.

1.2.1 Druhy dezénu

V závislosti na typu vzorku dezénu řadíme pneumatiky do následujících kategorií:

- **Symetrické**
- **Asymetrické**
- **Směrové**

1.2.1.1 Symetrický dezén

Symetrické pneumatiky jsou v dnešní době standardním typem používaným především u nižší a střední třídy vozidel. Hlavní výhodou jsou vyvážené vlastnosti na každém povrchu, fungují dobře jak na suché, tak i na mokré vozovce.

Díky symetrii dezénu lze pneumatiku umístit na vozidle na libovolnou pozici.



Obrázek 1.4: Typy dezénu (zleva symetrický, asymetrický a směrový)[3]

1.2.1.2 Asymetrický dezén

Asymetrické pneumatiky jsou rozděleny do dvou odlišných částí, vnitřní a vnější. Vnitřní strana dezénu slouží k odvodu vody a přenosu záběrových a brzdných sil na vozovku. Oproti tomu strana vnější zajišťuje vedení v přímém směru a stabilitu v zatáčkách.

Při montáži musíme dbát na označení na bočnicích pneumatiky, tzn. strana s označením „outside“ patří na vnější stranu a „inside“ na stranu vnitřní.

1.2.1.3 Směrový dezén

Směrové pneumatiky můžeme vnímat jako jistou formou symetrického dezénu. Jsou charakteristické tzv. šipovým designem, který má velmi dobrý záběr po celé ploše. Tento vzorek výborně odvádí sněh i vodu z běhounu, a proto je nejpoužívanější u zimních pneumatik.

Při montáži je třeba dbát na směr otáčení, proto má každá pneumatika na bočnici šipku. Toto označení udává směr ve kterém má být pneumatika namontována. [12]

1.3 Forenzní využití

Zkoumání otisků, ať už od pneumatik, bot či otisků prstů je nedílnou součástí současné kriminalistiky. V této sekci jsou popsány parametry, které jsou zkoumané na otisku pneumatiky z místa činu.

Otisk pneumatiky je výsledkem přenosu detailu pneumatiky na podklad.

V závislosti na podkladu rozlišujeme otisky na *3D* a *2D*. O 3D stopách mluvíme v případě otisknutí do půdy, písku či sněhu, oproti tomu 2D stopy vzniknou pokud pneumatika projede skrz prach, krev nebo jiný materiál.

Při odebírání otisků je nutné zaznamenat maximum informací jako šířku otisku, případně i hloubku otisku.

1.3.1 Charakteristiky otisku

Přenos vzorku, velikosti a dalších získaných rysů může sloužit jak k identifikaci podezřelého vozidla, tak i k eliminaci stop od vozidel záchranných složek. V této části, si podrobněji přiblížíme co vše může pomoci kriminalistům při identifikaci pneumatiky oproti danému otisku.

1.3.1.1 Design

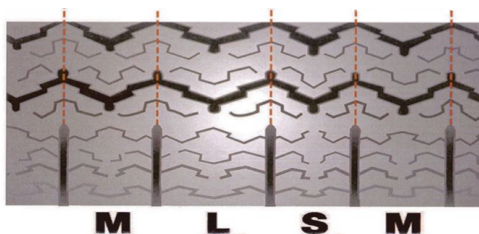
Design vzorku se skládá z velmi specifických a detailních uspořádání bloků, drážek a lamel. Některé vzorky mají podobný vzhled jako jiné, ale k závěru, že design odpovídá je potřeba, aby celá část vzorku viditelná na otisku byla totožná s otiskem zkoumané pneumatiky.[4]

1.3.1.2 Rozměr a vnější hlučnost

Většina pneumatik se vyrábí ve větším množství velikostí. Rozměr otisku ukazuje na fyzickou velikost běhounu, která může pomoci určit o jaký typ vozidla se jedná. Např. nákladní vozidla mají přirozeně pneumatiky širší než vozidla osobní.

Každá pneumatika je zařazena do určité **třídy vnější hlučnosti** podle množství hluku vydávaném za jízdy. Pro redukci tohoto hluku jsou po obvodu běhounu úseky různých velikostí. Na obrázku 1.5 je uveden příklad sekvence šířek úseků na pneumatice.

Pokud je z otisku zřetelná poloha **ukazatele opotřebení** v kombinaci s variací úseků, dostaneme tak další cennou informaci v procesu porovnávání.[4]



Obrázek 1.5: Příklad sekvence úseků pro snížení hlučnosti[4]

1.3.1.3 Znaký opotřebení

Třecí síly způsobují erozi na pryži pneumatiky, která ovlivňuje *viditelnost prvků* na blocích, lamelách a některých drážkách. Například nějaké lamely a dokonce i nějaké příčné drážky částečně zmizí.

Mnoho těchto znaků může být znatelných právě i na otisku. Tímto porovnáním se počet dalších možných pneumatik stejného rozměru a designu výrazně zredukuje.[4]

1.3.1.4 Individuální charakteristiky

Náhodné, individuální charakteristiky zahrnují *škrábance, řezy, praskliny a odřeniny* způsobené v průběhu užívání. Zároveň se jedná i o *objekty náhodně uvízlé ve vzorku*, jako jsou kamínky, sklo, hřebíky a další dočasně či trvale zachycené artefakty na běhounu.

Jejich velikost, tvar, orientace, stejně tak i přesná pozice v závislosti na jednotlivých prvcích pneumatiky je činí hodnotným přínosem při identifikaci pneumatiky.[4]

Analýza a návrh

Nalezení hledaného vzorku pneumatiky z databáze s mnoha záznamy v krátkém čase je kritické. Přestože bylo navrženo mnoho algoritmů k extrakci vzorků z obrazu, průzkum literatury ukazuje, že jen malé množství se specializuje právě na získávání vzorku z otisku pneumatiky.[5]

Na začátku této kapitoly jsou popsány algoritmy použitelné pro zpracování vzorku z fotografie a nalezení shody v databázi. Následně jsou rozebrána uživatelská rozhraní používaná v současnosti a v poslední části kapitoly je nastíněn návrh programu, který je v rámci práce vyvíjen.

2.1 Nástrahy zpracování

Základním kamenem pro všechny metody je databáze, ve které se nacházejí fotografie známých vzorků společně s technickými parametry. Tyto databáze byli dříve pouze ve formě knížek a všechno bylo potřeba hledat manuálně.

Díky pokroku technologií se však toto zdlouhavé manuální vyhledávání přenechává počítačům. Databáze mají podobnou podobu, s jedním rozdílem, jsou elektronické a umožňují vyhledávání podle parametrů.

Existuje několik různých přístupů, jeden založený na digitálním zpracování obrazu, jiný na schopnosti člověka rozpoznat a vyhodnotit jednotlivé prvky. Příklady konkrétních v současnosti používaných programů naleznete ke konci této kapitoly.

2.2 Algoritmy

Většina algoritmů pracuje s předzpracovanými obrázky, kde se díky předzpracování zvýrazní linie důležité pro porovnávání. Stejně tak i pro algoritmus z článku v ruském deníku o nedestruktivních metodách testování *"Identification of a Tire-Tread Pattern from Its Track Using a Computer-Aided Video*

Technique"[13] je nutné fotografii předzpracovat použitím binarizace² s daným prahem³.

Posléze se vybere z fotografie oblast, která obsahuje všechny hlavní identifikátory daného vzorku. Takto vybraná oblast je příkládána, bez změny orientace, na oblast stejných rozměrů do obrázku z databáze (se kterým testujeme shodu). Srovnávání probíhá pomocí techniky odečítání jednoho obrazu od druhého, a to pro všechna možná umístění. Jako nejpodobnější jsou tedy určena data s minimálním rozdílem.

Problémem nejen tohoto, ale i mnoha dalších algoritmů, je neflexibilita vůči rotaci obrazu. To je zapříčiněno tím, že algoritmus pracuje pouze s posunem obrazu, nikoliv s rotací. Je nezbytné, aby otisk byl ve stejné orientaci jako je obraz v databázi, protože v případě natočení hledaného obrazu, i o pouhých pár stupňů, nebude shoda nalezena.

Problémem identifikace vzorku nezávisle na jeho rotaci se zabývali v centru pro zpracování obrazu a informací na Xi'an univerzitě v článku „*Study on rotation-invariant texture feature extraction for tire pattern retrieval*“.[5]

Podrobněji se zde zabývali algoritmem **Radon-DTCWT** a algoritmem **CEDA**. Pro pochopení zmíněných algoritmů bude popsáno i několik základních technik, z kterých tyto algoritmy vycházejí.

DT-CWT

Metoda je účinná v přístupu k posunutým obrazům, kromě toho má také lepší směrovou selektivitu při filtrování vícerozměrných signálů.[14]

Souhrn vlastností DT-CWT:

- přiměřená tolerance vůči posunu
- dobrá směrová citlivost
- výborná rekonstrukce pomocí lineárních filtrů
- omezená redundance⁴ nezávislá na měřítku, 2 : 1 pro 1D (2^m : 1 pro m-D)
- efektivní výpočet N pořadí, 2-krát jednoduché DWT⁵ pro 1D (2^m -krát pro m-D)

Formální definice zní následovně:

$$\psi_c(x) = \psi_h(x) + j\psi_g(x) \quad (2.1)$$

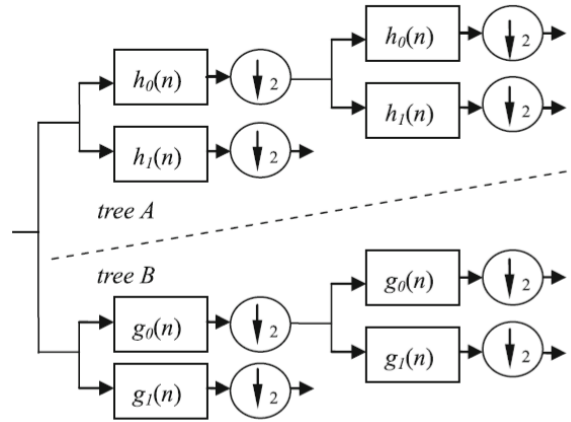
kde j je imaginární a ψ_h a ψ_g jsou ortogonální nebo biortogonální reálné vlnky.

²převedení obrazu do dvou binárních hodnot

³hranice určující zlom mezi hodnotami

⁴nadbytečnost

⁵discrete wavelet transform



Obrázek 2.1: Struktura 1D DT-CWT filtru[5]

Na obrázku 2.1 je ukázán filtr pro zpracování 1D signálu. Skládá se ze 2 reálných paralelních vlnkových stromů, strom A (také nazýván reálný strom) produkuje reálnou část koeficientů a strom B (imaginární strom) část imaginární.

DT-CWT pro funkci $f(x)$ v R^2 je definován vztahem:

$$\langle f, \psi_c \rangle = \langle f, \psi_h \rangle + j \langle f, \psi_g \rangle \quad (2.2)$$

a bude používán později v algoritmu Radon-DTCWT.

Radonová transformace

Radonová transformace počítá projekci obrazu ve specifickém směru. Projekce 2D funkce $f(x_1, x_2)$ je množina paralelních lineárních integrálů, pak funkce $f(x_1, x_2)$ v R^2 je definována vztahem:

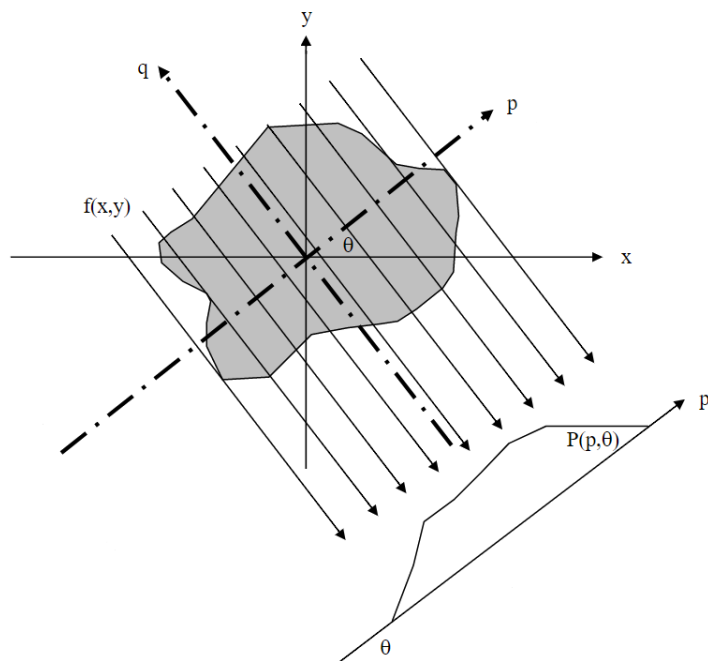
$$R_f(\theta, t) = \int_R \int_R f(x_1, x_2) \delta(x_1 \cos \theta + x_2 \sin \theta - t) dx_1 dx_2 \quad (2.3)$$

kde $\delta(\dots)$ je funkce nerovná 0 pouze pro argument rovný 0, θ určuje směr otočení a R_f je integrální projekce pro každé θ .

Pokud funkci originálního obrazu označíme $f_1(x_1, y_1)$ pak obraz po rotaci o $\Delta\theta$ označme $f_2(x_2, y_2)$ a platí:

$$\begin{aligned} R_{f_2}(\theta, t) &= \int_R \int_R f(y_1, y_2) \delta(y_1 \cos(\theta - \Delta\theta) + y_2 \sin(\theta - \Delta\theta) - t) dy_1 dy_2 \\ &= R_{f_1}((\theta - \Delta\theta), t) \end{aligned} \quad (2.4)$$

RT nabízí principy identifikace, které jsou neměnné v závislosti na rotaci a zrcadlení. Toho se dosáhne díky převedení souřadnic x a y na polární nebo semi-polární souřadnice.



Obrázek 2.2: Geometrické závislosti 2D Radonové transformace[6]

Klasifikace přímo ze sinografu⁶ má mnoho výhod, např. nezáleží na rotaci ani na případné zrcadlové projekci funkce. Rotace je znázorněna jako posun v ose x a zrcadlení je promítnuto jako invertovaná data na osu y . [15]

Z RT získáme 1D sekvenci pro každý úhel projekce. Tento algoritmus promítá obrázek do N orientací, kde každá má N podbodů (N je velikost obrázku). Projekční matice se tak skládá z N orientací.

$$R_f = \begin{bmatrix} g_{\theta_1}(1) & g_{\theta_1}(2) & \cdots & g_{\theta_1}(N) \\ g_{\theta_2}(1) & g_{\theta_2}(2) & \cdots & g_{\theta_2}(N) \\ \vdots & \vdots & & \vdots \\ g_{\theta_N}(1) & g_{\theta_N}(2) & \cdots & g_{\theta_N}(N) \end{bmatrix} \quad (2.5)$$

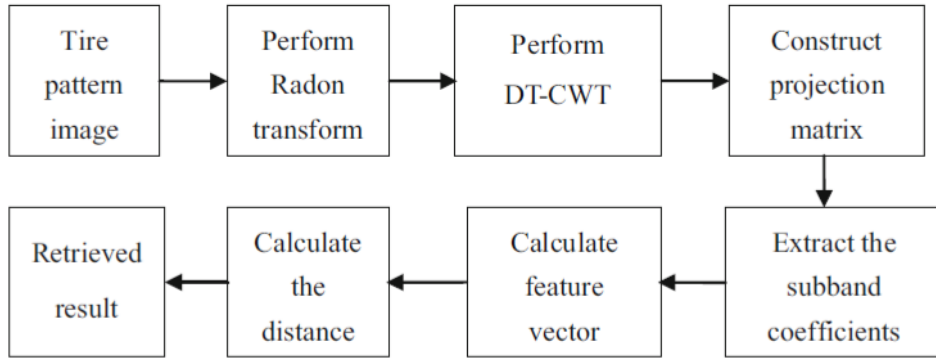
2.2.1 Radon-DTCWT

Jedná se o kombinaci radonové transformace společně s komplexní vlnkovou transformací (DT-CWT) a je definována jako:

$$\begin{aligned} DTRFT_f &= \langle R_f, \psi_c \rangle = \langle R_f, \psi_h \rangle + j \langle R_f, \psi_g \rangle \\ &= \int_{R^2} \psi_c(x) R_f(\theta, t) dt + j \int_{R^2} \psi_g(x) R_f(\theta, t) dt \end{aligned} \quad (2.6)$$

⁶prostor se semi-polárními souřadnicemi

Na diagramu 2.3 je znázorněn průběh tohoto algoritmu.



Obrázek 2.3: Vývojový diagram algoritmu Radon-DTCWT[5]

Jednotlivé kroky **Radon-DTCWT** algoritmu jsou následující:

1. Provedení radonové transformace u originálního obrázku, kde $(g_{\theta_i}(1), g_{\theta_i}(2), \dots, g_{\theta_i}(N))$ je sekvencí projekce $f(x, y)$ libovolného úhlu. Ze všech takto získaných hodnot je složena projekční matice R_f .
2. Na každý řádek matice R_f je provedena 3-úrovňové DT-CWT, z nichž každá se skládá z 9 vysokofrekvenčních subpásem a 1 nízkofrekvenčního subpásma.
3. Pro každé subpásma se spočítá střední hodnota (M), rozptyl (σ) a energie (E) pro každé i :

$$M_i = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N |I_i(x, y)| \quad (2.7)$$

$$\delta_i = \sqrt{\frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N (I_i(x, y) - M_i)^2} \quad (2.8)$$

$$E_i = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N I_i^2(x, y) \quad (2.9)$$

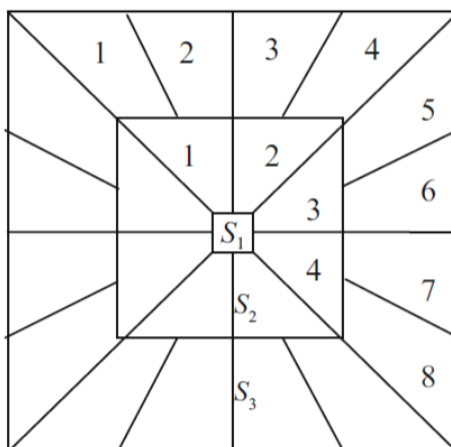
Vznikne tak příznakový vektor v následující podobě:

$$f = (M_1, \delta_1, E_1, M_2, \delta_2, E_2, \dots, M_{10}, \delta_{10}, E_{10})$$

4. Příznakový vektor se normalizuje a spočte se Euklidovská vzdálenost mezi dvěma obrázky. Touto vzdáleností se měří podobnost obrázků a jsou tak získány obrázky s nejmenší vzájemnou vzdáleností.

2.2.2 CEDA

Základním kamenem tohoto algoritmu je **Curvelet transformace**, která si rozdělí obrázek do subpásem s kterými následně pracuje. Rozdělení do úrovní a subpásem můžeme vidět na obrázku 2.4



Obrázek 2.4: Frekvenční mapa Curvelet transformace[5]

V závislosti na frekvenci jednotlivých subpásem transformovaného obrázku se určí **směrové charakteristiky vzorků pneumatik**. Vzorky jsou klasifikovány do 4 kategorií:

- **vertikální** – subpásma s největší energií je vertikální
- **horizontální** – subpásma s největší energií je horizontální
- **diagonální** – poměr energie vertikálního a horizontálního typu je podobný, největší energii má diagonální subpásma
- **nepravidelné** – pokud neodpovídá žádnému z předcházejících typů

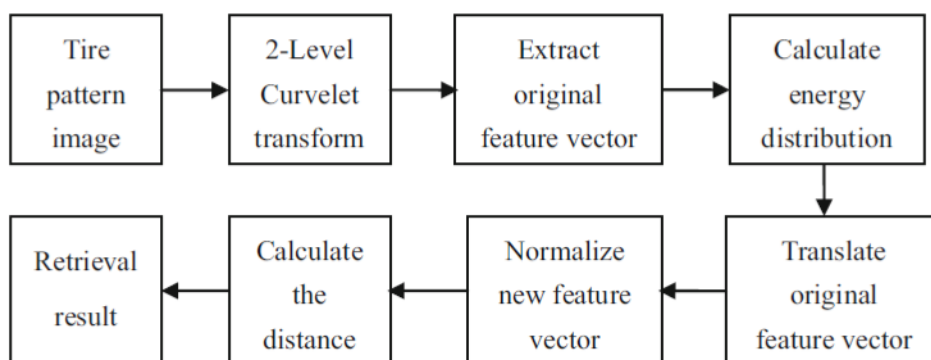


Obrázek 2.5: Druhy vzorků pneumatiky (zleva vertikální, horizontální, diagonální a nepravidelný)[5]

Pokud nastane u **Curvelet transformace** rotace obrázku, tak i koncentrace energie v subpásmu bude posunuta podle rotační osy. Tento posun může ovlivnit přesnost výsledku.

Aby se tomuto problému předcházelo, algoritmus **CEDA** rekonstruuje příznakový vektor a eliminuje tak vliv rotace. Pokud tedy obrázek vertikálního typu otočíme o 90° , otočený obrázek se změní na typ horizontální. Algoritmus může být popsán následujícími kroky:

1. Provedení 2-úrovňové Curvelet transformace obrázku vzorku pneumatiky.
2. Spočítá se střední hodnota a rozptyl pro 13 subpásem, které vytvoří příznakový vektor.
3. Spočítá se energie pro každou úroveň a subpásmo s největší energií je označeno jako referenční.
4. Kruhově se posouvá vektor příznaků, dokud se na první místo nedostane hodnota referenčního subpásma.



Obrázek 2.6: Vývojový diagram algoritmu CEDA[5]

Díky umístění subpásma s nejvyšší energií na prvním místě, nebude referenční subpásmo změněno vlivem rotace obrázku. Navíc vztahy a uspořádání mezi komponenty vektoru nejsou narušeny, což udržuje singularitu příznakového vektoru.

2.3 Existující rozhraní

Tato sekce popisuje reálná rozhraní, která se na dnešním trhu vyskytují. Bohužel se většinou jedná již o hotový produkt určený k prodeji a tak není možné dohledat algoritmy na kterých je program vybudován.

2. ANALÝZA A NÁVRH

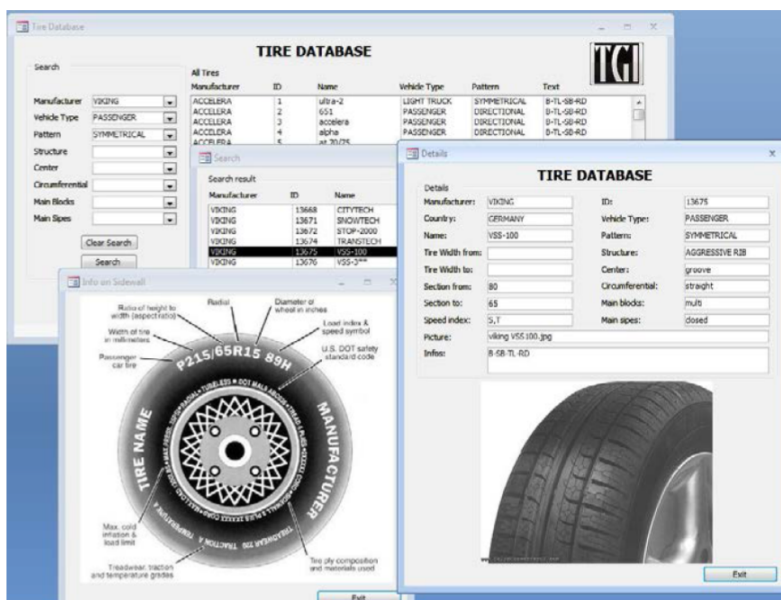
2.3.1 „Databáze pneumatik“

Tato databáze obsahuje vzorky všech světových značek, což je nezbytné, neboť cca 60% pneumatik je do USA dováženo z Evropy a Asie. Záznamy obsahují fotografie pneumatik i jejich otisků na nejrůznějších površích, některé dokonce i na lidské pokožce. V roce 2013 bylo v databázi 15 838 vzorků od 357 různých společností vyrábějících pneumatiky.

Pro snadnější použití celosvětově, je databáze ve formátu Microsoft Excel[©] a Microsoft Access[©], náhled obou variant můžete vidět na obrázcích 2.7 a 2.8.

| COMPANY INFO | | TIRE INFO | SIZE INFO | DATABASE-INFO | SEARCH |
|--------------|-----------|-----------|-----------|------------------|------------|
| MANUFACTURER | COUNTRY | D | NAME | Vehicle Type | Pattern |
| 4 ACCELERA | INDONESIA | 1 | 651 | 65 50 V 2010 | B-TL-SB-RD |
| 5 ACCELERA | INDONESIA | 2 | accelera | 70 65 H 2010 | B-TL-SB-RD |
| 6 ACCELERA | INDONESIA | 3 | alpha | 65 45 V W 2010 | B-TL-SB-RD |
| 7 ACCELERA | INDONESIA | 4 | alpha | 75 70 S 2011 | B-TL-SB-RD |
| 8 ACCELERA | INDONESIA | 5 | delta | 70 50 H V W 2010 | B-TL-SB-RD |
| 9 ACCELERA | INDONESIA | 6 | delta | 45 45 W 2010 | B-TL-SB-RD |
| 10 ACCELERA | INDONESIA | 7 | epsilon | 65 60 H V 2010 | B-TL-SB-RD |
| 11 ACCELERA | INDONESIA | 8 | gamma | 65 55 T H 2010 | B-TL-SB-RD |
| 12 ACCELERA | INDONESIA | 9 | iota | 65 50 V 2010 | B-TL-SB-RD |
| 13 ACCELERA | INDONESIA | 10 | phi | 50 30 Y 2010 | B-TL-SB-RD |
| 14 ACCELERA | INDONESIA | 11 | phi_2 | 30 Y 2011 | B-TL-SB-RD |
| 15 ACCELERA | INDONESIA | 12 | rho | 70 60 H V 2010 | B-TL-SB-RD |
| 16 ACCELERA | INDONESIA | 13 | sigma | 35 35 W 2010 | B-TL-SB-RD |
| 17 ACCELERA | INDONESIA | 14 | snow | 2011 | B-TL-SB-RD |
| 18 ACCELERA | INDONESIA | 15 | ultra-2 | 82 P 2011 | B-TL-SB-RD |
| 19 ACCELERA | INDONESIA | 16 | zeta | 65 40 T H V 2011 | B-TL-SB-RD |
| 20 ACHILLES | USA | 17 | lab sport | 80 30 V W 2012 | B-TL-SB-RD |
| 21 ACHILLES | USA | 18 | platinum | 70 60 H 2012 | B-TL-SB-RD |

Obrázek 2.7: Databáze v EXCEL formátu[7]



Obrázek 2.8: Databáze v ACCESS formátu[7]

V databázi jsou k nalezení i otisky kol motocyklů. Vyhledávání lze realizovat pomocí jednoho ze dvou kritérií, podle jména vzorku či jména výrobce nebo s použitím 7 charakteristik vzorku.

- **Typ vozidla** - osobní, nákladní, přípojný, ...
- **Charakteristika vzorku** - směrový, symetrický, asymetrický
- **Struktura vzorku** - sněhová pneumatiky, letní pneumatika, ...
- **Střed vzorku** - drážka, žebro, bloky, ...
- **Obvodové drážky ve středu pneumatiky** - přímé, zigzag, ...
- **Tvar bloků** - čtverec, V-tvar, ...
- **Tvar lamel uvnitř bloku** - přímé, zakřivené, ...

Použití

Nejprve se musí klasifikovat specifické vlastnosti vzorku, což provádí operátor⁷. Operátor vyplní co nejvíce ze 7 charakteristik, které je schopný rozpoznat z fotografie a vloží do systému. Databáze navrátí něco mezi 1 a 15 000 záznamy, většinou ne více než 10. Pro tyto výsledky je potřeba, aby operátor provedl vizuální porovnání a našel tak shodu.

Tato databáze je používána i rozhraním **TreadMate**, kterým se zabývá následující podkapitola.

2.3.2 TreadMate

Jedná se o kolekci pneumatik udržovanou stejnou společností z Velké Británie, jako podobný systém jménem SoleMate⁸. V této databázi od **Foster & Freeman USA Inc.** se nachází informace o více než 8 500 pneumatikách a jejich vzorcích. Může být používána samostatně nebo s využitím systému SICAR (obr.2.9), evidenčního systému pro otisky od bot a pneumatik vyvinuté stejnou společností.[8]

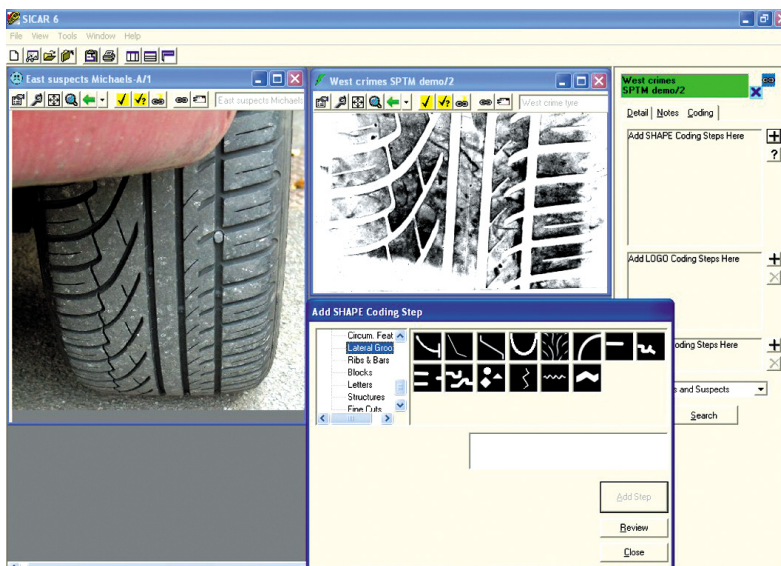
U každého záznamu je evidováno:

- výrobce
- datum uvedení na trh
- ukázka dezénu (fotografie)
- zakódovaný set charakteristických prvků na vzorku

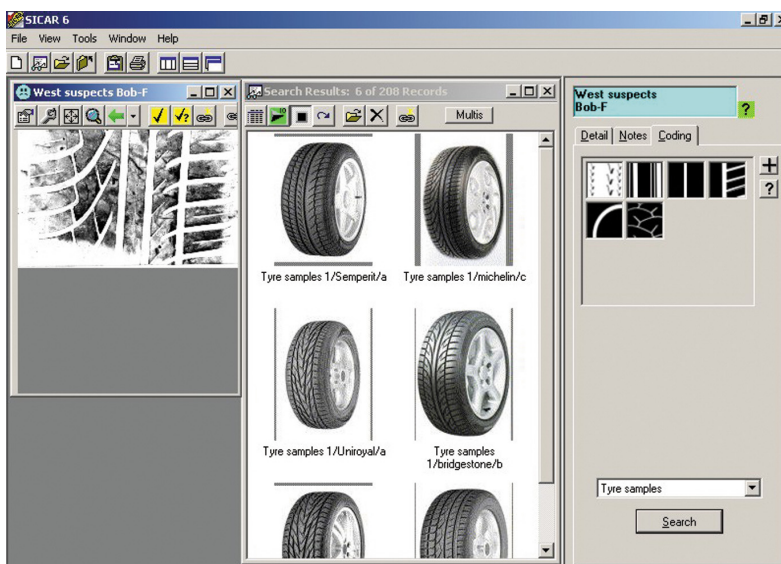
⁷ osoba obsluhující systém

⁸ databáze otisků podrážek bot

2. ANALÝZA A NÁVRH



Obrázek 2.9: Ukázka GUI TreadMate[8]



Obrázek 2.10: Ukázka nalezených pneumatik[8]

Protože výrobci někdy používají stejný design běhounu, může být nalezení přesné shody obtížné. Pro tyto případy jsou záznamy s takovou shodou propojeny, tedy při hledání kritického otisku jsou tyto skutečnosti uvažovány.

Použití

Identifikace základních prvků probíhá operátorem, kde prvky jsou znázorněny graficky a operátor určuje nejlepší vizuální shodu.

V systému jsou tyto prvky zakódovány a použity jako kritéria pro hledání nejlepší shody. Výsledky jsou opět zobrazeny operátorovi, v klesajícím pořadí podle míry shody, výběr odpovídajícího exempláře závisí opět na vizuálním porovnání operátora (obr.2.10).

2.3.3 Studie (Wang 2005)

Poslední rozhraní se od předcházejících výrazně liší, jako jediné pracuje s šířkou drážek a žebel. Jedná se o program vyvíjený v rámci studie profesora Wanga „*A tire-mark identification scheme for suspected vehicle detection in hit and run accident*“.[9]

Autorka se snaží zrychlit a zpřesnit dosažení výsledků při identifikaci vozidla podle otisků pneumatik z místa nehody. Navrhuje prvky, které slouží jako identifikační znaky při porovnávání. Jím navržená struktura databáze se dá dělit na 3 hlavní pod-databáze:

- databáze pneumatik
- databáze vozidel
- základní informace o výrobci

Na diagramu 2.11 je vyobrazená úplná struktura databáze včetně obsahu a parametrů, podle kterých se vyhledává shoda.

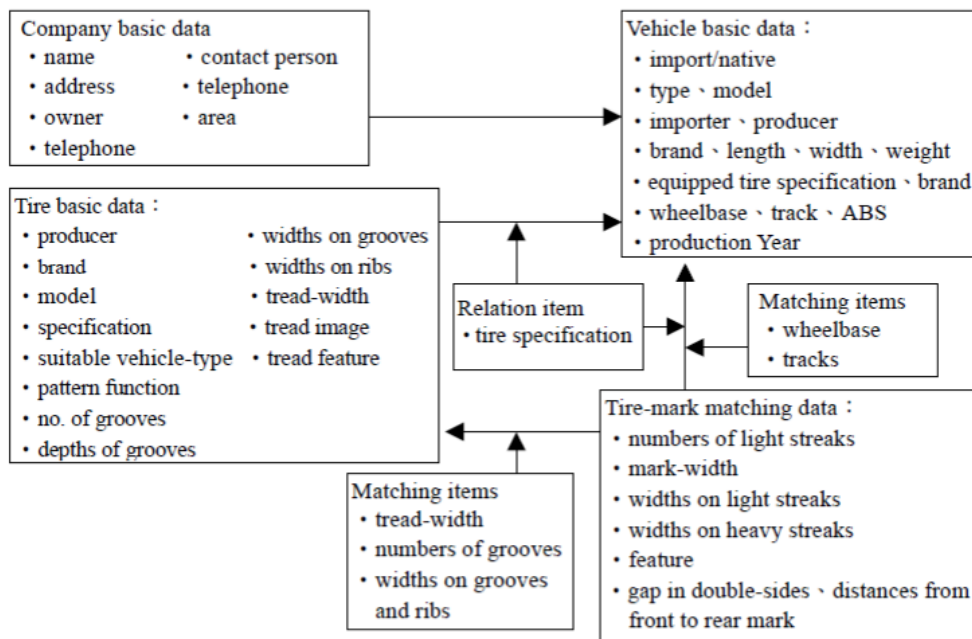
Databáze neslouží pouze k vyhledávání podle vzorku pneumatiky, ale i podle jiných parametrů (např. rozteč kol). Ty však v této práci nebudeme uvažovat.

Použití

Použití se výrazně neliší od předchozích, i u tohoto prototypu se musí data zadat manuálně. Operátor pracuje s rozhraním, které můžete vidět na obrázku 2.12. Po vyplnění známých hodnot přijde na řadu implementovaný srovnávací algoritmus. Systém vrátí několik záznamů s vyčísleným stupněm shody⁹, ale konečné vyhodnocení je opět na operátorovi.

⁹pravděpodobnost shody

2. ANALÝZA A NÁVRH



Obrázek 2.11: Struktura a obsah databáze[9]

The screenshot shows a software interface for tire tread analysis. The window title is "Tread-potenz. setbird. systm".

Input fields:

- Input tire mark width?(cm): 14.5
- Input number of light streaks?: 3

Table of light-streak width:

| width | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-------|------|------|------|------|------|------|------|------|
| | 1.00 | 0.67 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

Table of heavy-streak width:

| width | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|-------|------|------|------|------|------|------|------|
| | 2.33 | 2.33 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

Other parameters:

- tire-tread deformation: -0.05 cm
- parameter: 0.50
- matching degree: 95 %
- front dis.: 1.45 m
- rear dis.: 1.45 m
- wheelbase: 0.00 m

Buttons: "Inquiry Tire Tread", "PREV. PAGE", "FIND BELONGING".

Status bar: Total: 10 Records found

Obrázek 2.12: Rozhraní databáze[9]

2.4 Návrh

Jako první krok při navrhování aplikace je potřeba ujasnit si, co všechno bude třeba pro její bezproblémový chod. V případě této aplikace se neobejdeme bez 3 hlavních komponent.

- **Databáze** - je potřeba uvažovat nad reprezentací dat v programu, tak i o databázi zobrazované uživateli
- **Uživatelské rozhraní** - pro interakci s uživatelem je potřeba vyvinout jednoduché a intuitivní rozhraní, kterým bude aplikaci ovládat
- **Vnitřní funkce** - aplikace musí obsahovat funkce a algoritmy (zdrojový kód), které se postarají o její funkcionalitu

Návrh čerpá ze znalostí nabitých při zpracování rešeršní části kombinovaných se specifickým zadáním této práce.

Aplikace bude závislá na znalosti a rozlišovacích znalostech operátora, stejně jako u předchozích, bude třeba znát základy ohledně prvků vyskytujících se na běhounu pneumatiky (kapitola č. 1). Také se využije schopnosti člověka vizuálně porovnávat různé objekty.

2.4.1 Databáze

Vytvoření databáze je komplikovaný úkol, protože fotografie a specifikace jednotlivých pneumatik je nutné dohledávat jednu po druhé. Ve formě dat vhodných pro přímé vložení do databáze nejsou volně přístupné. Proto bude databáze skromnějších rozměrů, a to sice v desítkách záznamů, nikoliv v tisících, jak je tomu běžné u reálně používaných systémů.

K vytvoření vnitřní databáze se použije stejné uživatelské rozhraní, které se následně použije pro vyhledávání shody v databázi. Toto bude umožněno díky *průběžnému ukládání dat* při zadávání hledaného otisku. Po zadání všech viditelných parametrů operátorem bude tedy namísto spuštění vyhledávání, záznam přidán do databáze na úrovni programu.

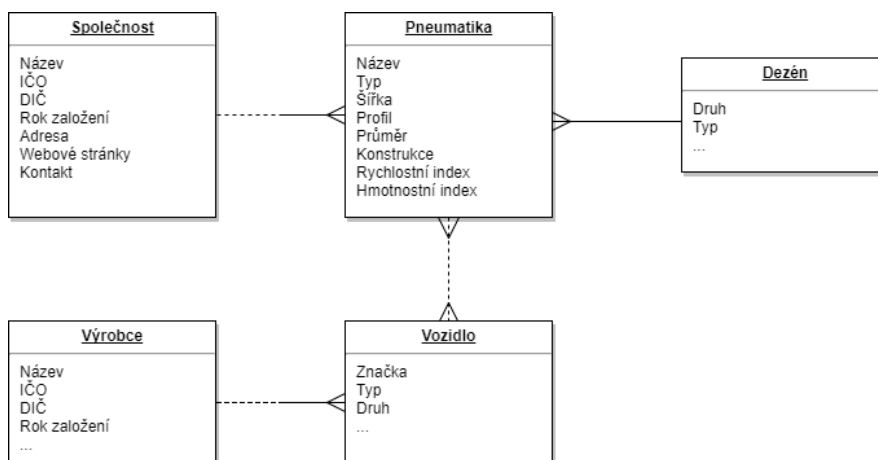
Tato funkce nebude však volně přístupnou pro uživatele aplikace. Což bude sloužit jako opatření proti neodbornému zásahu do databáze, které by mohlo nastat při přidávání záznamů nedostatečně kvalifikovanou osobou.

Databáze ve formě zobrazované uživateli musí být vytvořena separátně, neboť z fotografie vzorku nelze přesně zjistit ani výrobce, ani upřesňující parametry pneumatiky. Na diagramu 2.13 můžeme nahlédnout na navrhovaný *konceptuální model* této databáze.

2.4.1.1 Reprezentace v paměti

Reprezentace databáze v paměti programu je dalším *kritickým bodem*. Aby bylo vyhledávání co nejefektivnější je potřeba zvolit správnou reprezentaci

2. ANALÝZA A NÁVRH



Obrázek 2.13: Konceptuální model databáze

dat, proto autorka nejdříve uvažuje časovou složitost operací pro daný formát. V úvahu přicházejí autorce na mysl dvě možnosti - uložení ve formátu **XML** nebo využití **maticové reprezentace** rovnou v MATLABu.

Inicializace databáze je jednorázová činnost, která se nebude vícekrát opakovat, a proto není rozhodujícím kritériem. Pro celistvost je však uvedeno jakým způsobem se v MATLABu s XML pracuje.

Načtení XML dokumentu se provádí pomocí funkce `xmlread`, která převede obsah dokumentu do formátu **DOM**¹⁰, kde je každý záznam uložen do samostatného uzlu s danými atributy.

Přístup k jednotlivým záznamům je oproti načítání databáze kritické. Zde je více než vhodné udržet časovou složitost co nejnižší. Z XML dokumentu jsou uloženy záznamy v DOM pomocí nepojmenovaných struktur, což nedovoluje přístup právě k jedné vybrané struktuře. Za předpokladu, že lze vybírat struktury podle jména, bylo by stejně potřeba jména porovnávat jedno po druhém, dokud se nenalezne shoda. V nejhorším případě by tak byla složitost *přístupu k prvku* $O(n)$, kde n je počet záznamů.

Oproti tomu uložení v matici dovoluje přístup k prvkům pomocí indexace, díky čemuž je složitost přístupu ke konkrétnímu prvku konstanta $O(1)$. Nevýhodou zde zůstává přehlednost dat, která jsou zakódovaná v matici pomocí čísel. Ačkoliv může takovéto řešení působit poněkud zmatečně, v reálné situaci k nim nebude mít běžný uživatel přístup. S těmito daty bude operovat pouze vývojář pro něž to není komplikací.

Z výše zmíněných důvodů se autorka rozhodla využít **maticovou reprezentaci dat**. Navíc vývojové prostředí MATLAB je pro práci s maticemi přímo stvořené.

¹⁰Document Object Model[16]

| Pozice | Význam | Možnosti |
|--------|-----------------------------|--|
| 1 | šířka pneumatiky [mm] | 1 ... n |
| 2 | střed pneumatiky | 0 - nespecifikováno 1 - drážka 2 - žebro 3 - bloky |
| 3 | tvar obv.drážek ve středu | 0 - nespecifikováno 1 - přímé 2 - lomené 3 - obecné |
| 4 | počet obv. drážek ve středu | 0 ... n |
| 5 | počet přímých drážek | 0 ... n |
| 6 | počet lomených drážek | 0 ... n |
| 7 | počet obecných drážek | 0 ... n |
| 8 | počet bloků | 0 ... n |
| 9 | výskyt lamel | 0 - ne 1 - ano |
| 10 | počet hranatých bloků | 0 ... n |
| 11 | počet zaoblených bloků | 0 ... n |

Tabulka 2.1: Popis jednotlivých pozic vektoru

Každý záznam bude mít formu **vektoru** o 11 složkách. Jednotlivé složky vektoru jsou znázorněny v tabulce 2.1 včetně všech možných variant.

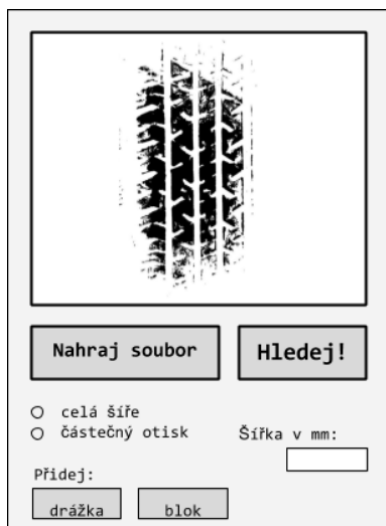
2.4.2 Uživatelské rozhraní

Uživatelské rozhraní musí umožnit načtení fotografie a grafické zaznamenávání prvků do této fotografie. K nahrání bude sloužit tlačítko (*Push Button*). Po jeho stisknutí se zavolá funkce `imgetfile()`. Při chybném vstupu aplikace zareaguje chybovou hláškou.

Další prvek potřebný v GUI je možnost zadat **reálnou šířku** otisku a vybrat, zda se jedná o **otisk celé šíře** pneumatiky nebo jen **otisk částečný**. Toho bude docíleno pomocí nabídky s možností vybrat pouze jednu z variant (*Radio Button*) a na základě této volby se stanoví algoritmus, který se použije pro vyhledávání. Následně bude potřeba zadat rozměr otisku do připraveného pole (*Edit Text*) a rozměr uložit.

V tuto chvíli bude dovoleno začít graficky zvýrazňovat identifikační prvky otisku. Výběrem a stisknutím tlačítka začne uživatel zadávat konkrétní prvek. Uživatel se tak dostane do **módu zadávání**, který bude ukončen až po stisknutím konkrétní klávesy z klávesnice. Ukončit režim úprav bude možné stisknutím jedné ze dvou kláves:

- **Enter** (ASCII 03) - potvrdí konec zadávání jednoho prvku (drážky, lamely, bloku, ...)
- **Esc** (ASCII 27) - ukončí zadávání lamel bloku



Obrázek 2.14: Návrh GUI

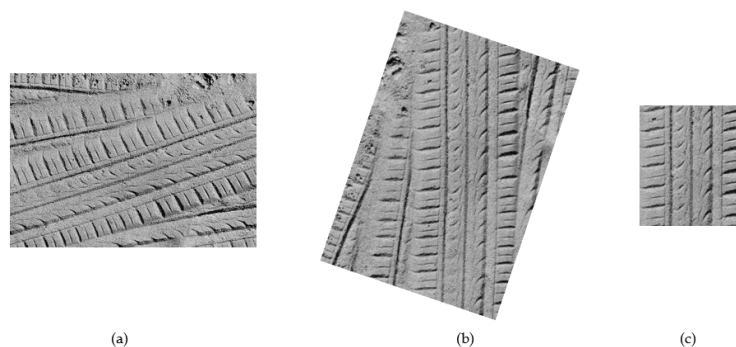
Pro rozlišení různých typů křivek - přímk, lomených čar a obecných křivek - budou autorkou implementovány různé funkce tvorby křivek pro různá tlačítka myši.

- **Levé tlačítko** slouží pro zaznamenávání bodů, které definují přímky mezi těmito body, pokud se do posloupnosti klikání levým tlačítkem nevloudí stisknutí pravého tlačítka myši.
- **Pravé tlačítko** indikuje zadávání obecné křivky. Pokud je detekován stisk pravého tlačítka, rozhraní nevykreslí spojnicí bodů, ale čeká na zadání koncového bodu (levé tlačítko myši).

2.4.2.1 Předzpracování dat

Aby bylo dosaženo správné funkce, je potřeba fotografii před nahráním do aplikace nejprve předzpracovat. Předzpracováním nejsou myšleny úpravy na úrovni změny struktury obrazu jako je prahování, detekce hran či jiná metoda používaná na zpracování digitálního obrazu.

Vstup programu není ošetřen vůči **rotaci** vyhledávaného vzorku, proto je zapotřebí si obrázek připravit do požadované polohy. Otisk musí být vždy orientován tak, že šířka pneumatiky je souběžná s vodorovnou osou. Na směřování vzorku u směrových pneumatik nezáleží, algoritmus je vůči této rotaci imunní.



Obrázek 2.15: Předzpracování: (a) originál, (b) po rotaci, (c) po vyříznutí

Druhý krok předzpracování je **oříznutí** obrázku tak, aby šíře obrázku odpovídala šířce otisku pneumatiky. Hlavním důvodem tohoto kroku je umožnit přepočítání ze vzdálenosti na fotografii do reálné vzdálenosti v milimetrech. Přepočítání se bude provádět na základě velikosti obrázku a zadané šířky od uživatele. Ukázka předzpracování je na obrázku 2.15.

2.4.3 Vyhledávání a zobrazení výsledků

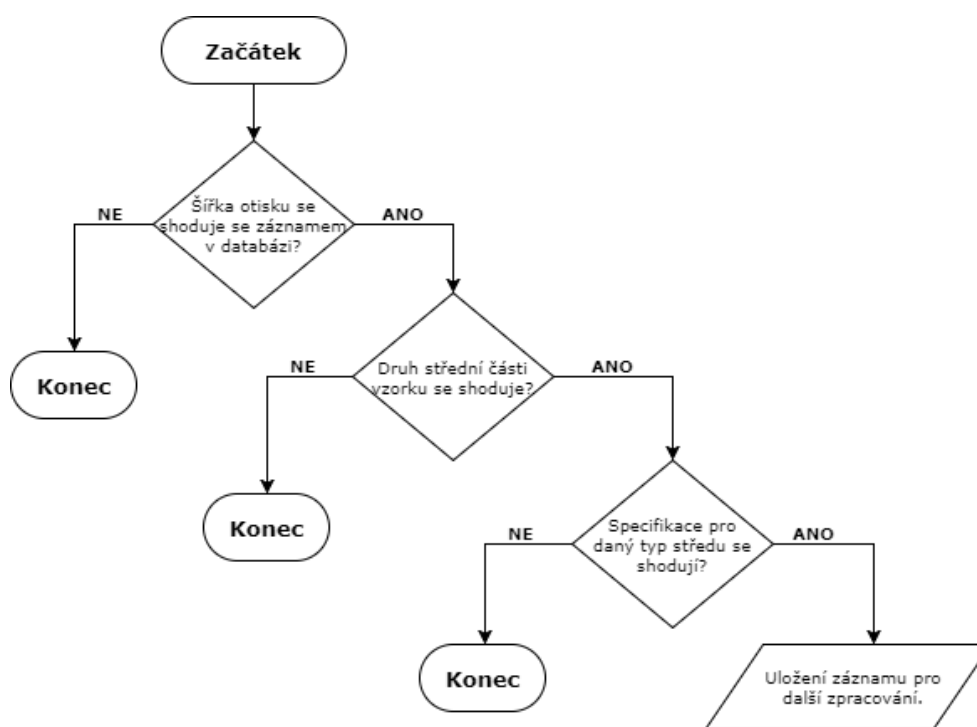
Vyhledávání bude implementováno odlišně od v praxi používaných systémů, protože již vnitřní databáze je koncipována odlišně. U tohoto druhu programů je důležité pracovat s přesnými hodnotami jako je např. šířka jedné drážky, teprve pak je možné získat dostatečně průkazné výsledky k přesné identifikaci jednoho konkrétního vozidla.

Bohužel tento postup není moc dobře realizovatelný pokud je databáze tvořená pouze pro testování jedné aplikace. Navíc jak již bylo uvedeno dříve, pro práci takového rozsahu není možné získat data takovéto přesnosti. Proto se celá aplikace zaměřuje na **porovnávání různých tvarů** bez ohledu na jejich velikost. Upřednostňuje se vizuální podoba otisku s pneumatikou před, na desetinu přesnými, rozměry jednotlivých drážek.

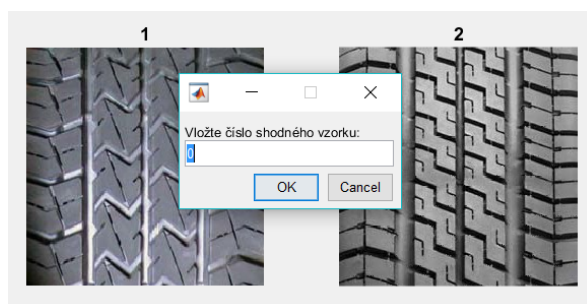
Jak již bylo řečeno, vyhledávací algoritmus bude rozdílný pro částečný a celý otisk. U celého otisku jsou jako hlavní parametry zvoleny **šířka otisku** (pneumatiky) a **definice středu běhounu**. Hledaný otisk se musí shodovat ve všech těchto prvcích, aby byl vůbec uvažován do dalšího porovnávání. Jednoduchý princip lze vidět na diagramu 2.16, kde lze také nahlédnout jaký parametr má jakou prioritu.

Záznamy, které splňují základní kritéria budou porovnávány parametr po parametru, čímž se určí nakolik si odpovídají. Po průchodu celou databází bude zjištěn maximální počet shodných parametrů. Záznamy s tímto počtem shodných parametrů budou zobrazeny uživateli v novém okně za účelem **vizuálního porovnání**.

2. ANALÝZA A NÁVRH



Obrázek 2.16: Algoritmus pro celý otisk



Obrázek 2.17: Volba shody

Výběr odpovídajícího vzorku bude realizován pomocí pop-up okna, do kterého uživatel zadá číslo příslušného vzorku. Pro případ absence shodného vzorku v databázi je ve vyplňovacím poli přednastavena hodnota 0, kterou uživatel pouze potvrdí stisknutím tlačítka *OK* (obr. 2.17).

Na základě tohoto výběru se zavolá záznam z databáze a zobrazí se uživateli (obr. 3.2).

Vyhodnocení **částečného otisku** není pevně svázáno s konkrétními parametry, protože není známo o jakou část otisku se jedná. Nelze proto použít informace o umístění jednotlivých prvků v rámci otisku a ani jejich počet. Tyto skutečnosti situaci značně komplikují, a proto při vyhledávání částečných otisků nebudou výsledky tak průkazné.

Vyhledávání pracuje na stejném principu jako vyhledávání u celého otisku, s tím rozdílem, že se porovnávají pouze relevantní parametry pro částečný otisk. Zobrazení výsledků je totožné jako u varianty s celým otiskem.

Realizace

V této kapitole se shrne a případně bude dodefinováno, jakým způsobem aplikace vznikla. Celá realizace je provedena ve vývojovém prostředí MATLAB, které nabízí mnoho možností a usnadňuje tak celý proces. Snaží se zasvětit čtenáře do konkrétních prvků MATLABu, které byly použity. Nechybí zde ani ukázky kódů ze zdrojových souborů aplikace.

3.1 Databáze

Vnitřní databáze byla, jak již bylo předesláno, vytvořena pomocí rozhraní aplikace. Postupuje se stejně jako v případě standardního použití aplikace, rozdílné je jen naložení se zaznamenanými hodnotami.

Jednotlivé záznamy ve formě vektorů se přidávají do databáze, namísto hledání shody v databázi. Databáze je reprezentována **maticí** o rozměru $n \times 11$, kde n je počet záznamů databáze.



Obrázek 3.1: Označené charakteristiky ve vyvíjené aplikaci

3. REALIZACE

Podle stanovených pravidel v tabulce 2.1, vzorek z obrázku 3.1 bude ve vnitřní reprezentaci vypadat následovně:

```
record = [225 3 0 2 0 2 0 3 1 3 0]
```

Verze uživatelské databáze je vytvořena diametrálně odlišně. Jednotlivé záznamy musí být přehledné a snadno čitelné pro uživatele, a proto je mimo jiné součástí záznamu i fotografie daného vzorku.

Položky evidované pro každou pneumatiku jsou následující:

- Název společnosti výrobce pneumatiky
- Název vzorku
- Druh vzorku (letní, zimní nebo univerzální)
- Typ vozidla
- Šířka pneumatiky v milimetrech
- Profil
- Průměr
- Druh konstrukce
- Hmotnostní index
- Rychlostní index
- Typ dezénu

Nejprve bylo potřeba najít vhodné obrázky vzorků pneumatik, ze kterých by bylo možné vytvořit databázi. Snímky vzorku pneumatiky by měly být fotografovány kolmo k pneumatice.

Použito je **přes 40 fotografií** z ukázkové knihovny vzorků pneumatik [17]. K těmto fotografiím bylo zapotřebí dohledat příslušné informace vhodné do uživatelské verze databáze. Tyto informace byly hledány v katalozích výrobců pneumatik na internetu, kde bylo nejprve nutné nalézt pneumatiku se stejným dezénem. Až v tuto chvíli se mohly do databáze **ručně vložit** nalezená data.

Vzhledem k takto časově náročným úkonům při tvorbě databáze, je databáze zjednodušená oproti původnímu návrhu. Jedná se pouze o **textové záznamy** bez prolinkování¹¹ s údaji o společnosti, výrobcu automobilu a dalších. Ukázkou jednoho takového záznamu naleznete na obrázku 3.2.

¹¹vzájemné propojení pomocí odkazů

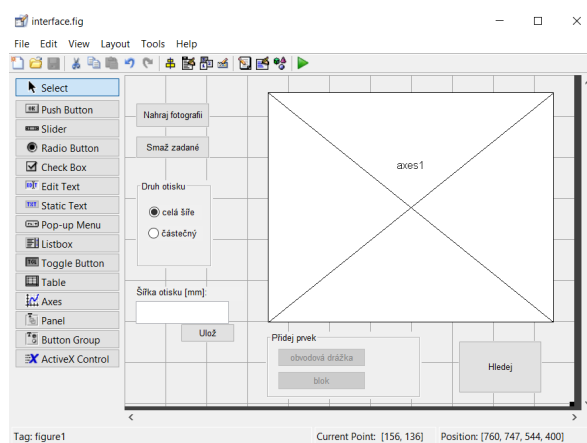
záznam č. 15
výrobce: Dunlop
název: Grandtrek ST30
druh: univerzální
typ vozidla: Offroad, SUV, 4x4
šířka: 225
profil: 60
průměr: 18"
konstrukce: R
hmotnostní index: 100
rychlostní index: H (210km/h)
dezén: směrový



Obrázek 3.2: Ukázka záznamu z databáze

3.2 Implementace

Veškerá implementace je realizována v jazyce MATLAB, dokonce i GUI je vytvořeno pomocí prostředí GUIDE[18], které MATLAB nabízí. GUIDE poskytuje možnost grafického vytvoření uživatelského rozhraní, které následně automaticky vygeneruje pro tento návrh MATLAB kód. Tento před-připravený kód grafického rozhraní lze modifikovat tak, aby se docílilo očekávaného chování aplikace.



Obrázek 3.3: Ukázka práce s GUIDE

3.2.1 Implementované funkce

K vykreslování prvků do otisku se používá upravená knihovní funkce MATLABu `ginput`[19]. Tato funkce není dostatečná pro navrhovaný prototyp, proto již jako odrazový můstek byla použita přizpůsobená verze této funkce `ginputc`[20].

3. REALIZACE

Tato funkce dovoluje programátorovi ovládat nejen tloušťku, styl a barvu vykreslované čáry, ale i mnoho dalších prvků.

Nicméně ani funkce `ginputc` nesplňuje všechny požadavky, proto autorka provedla potřebné změny v kódu. Konkrétně se jedná o přidání vstupního parametru `Block`, indikujícího zda se má první a poslední vyznačený bod spojit. Dále byla kvůli zadávání bloků společně s lamelami upravena podmínka pro ukončení zadávání.

Asi nejdůležitější obměnou kódu však zůstává přidání funkce na vykreslování obecných křivek `showSpline`. Tato funkce křivky jenom vykresluje, vstupní data jsou zpracována v nadřazené funkci.

```
function [] = showSpline( x, y, color, linewidth)
    t = 1:length(x);
    tt = linspace(t(1), t(end));
    xx = spline(t, x, tt);
    yy = spline(t, y, tt);
    plot(xx, yy, color, 'LineWidth', linewidth);
end
```

Detekování kliknutí *pravého tlačítka myši* se považuje za impuls pro vykreslení křivky pomocí funkce `showSpline`. Díky této funkci je operátor schopen zadat nejenom přímé křivky, ale i křivky obecné.

```
function [chPrms, grC, grS] = addGroove(fPrms, I, oneMm)
```

Funkce slouží k označování *obvodových drážek* v GUI, rovnou si i některé údaje zpracovává. Zaznamenává styl jednotlivých drážek, jejich umístění a samozřejmě i jejich počet.

```
function [changeParams] = addBlock(finParams, I, oneMm)
```

Funkce slouží k zadávání *bloků* v GUI a k zpracování jejich parametrů. Určuje zda se jedná o blok přímých či zaoblených tvarů, umístění i počet stejně jako předcházející funkce.

```
function [] = addSipe()
```

Funkce je volána z funkce `addBlock`, kterou tak doplňuje. Znázorňuje zadávání *lamel* v bloku (ze kterého je volána) a ukládá informaci o jejich počtu a tvaru.

```
function [halfX, a, b] = getCenter(img)
```

Funkce spočítá mezní hodnoty *středového pásma* ze vstupního obrázku, které následně i vrátí. Funkce umožňuje i zobrazení pomocných čar - okrajové hodnoty středového pásma.

function [minX, maxX] = getMinMax(x)

Jak již název napovídá, jedná se o funkci hledající *minimum a maximum* v daném poli. Používá se při lokalizování určitých prvků (v jakém rozmezí se nachází), většinou využito pouze pro *x-ové* souřadnice.

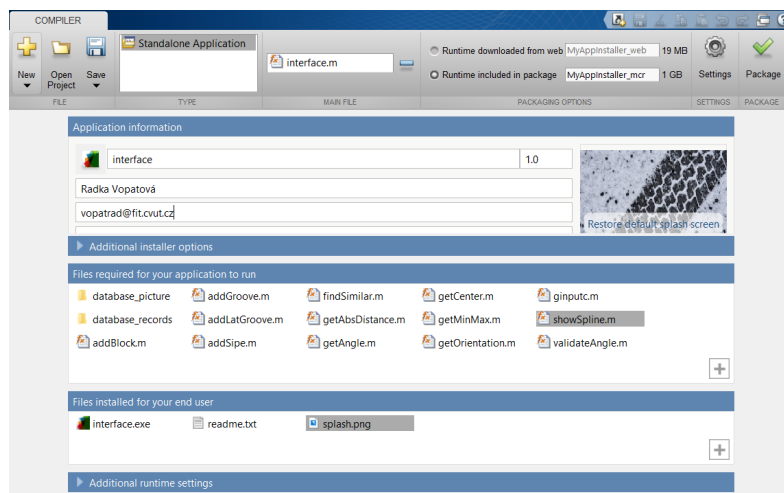
function [] = findSimilar(finParams)

Hlavní funkce, která má na starost konečné zpracování načtených dat a *nalezení nejpodobnějších záznamů* v databázi. Nalezené vzorky zobrazí a umožní operátorovi vybrat odpovídající vzorek. K tomuto vzorku pak vyhledá záznam v databázi a zobrazí ho.

3.2.2 Tvorba samostatně spustitelné aplikace

Opět je využito toho, co MATLAB nabízí. Přímo pomocí editoru lze vytvořit aplikace spustitelná mimo prostředí MATLAB. Toho je dosaženo pomocí *Application Compileru*[21] v podobě grafického rozhraní nebo pomocí příkazu `applicationCompiler`[22] v příkazové řádce.

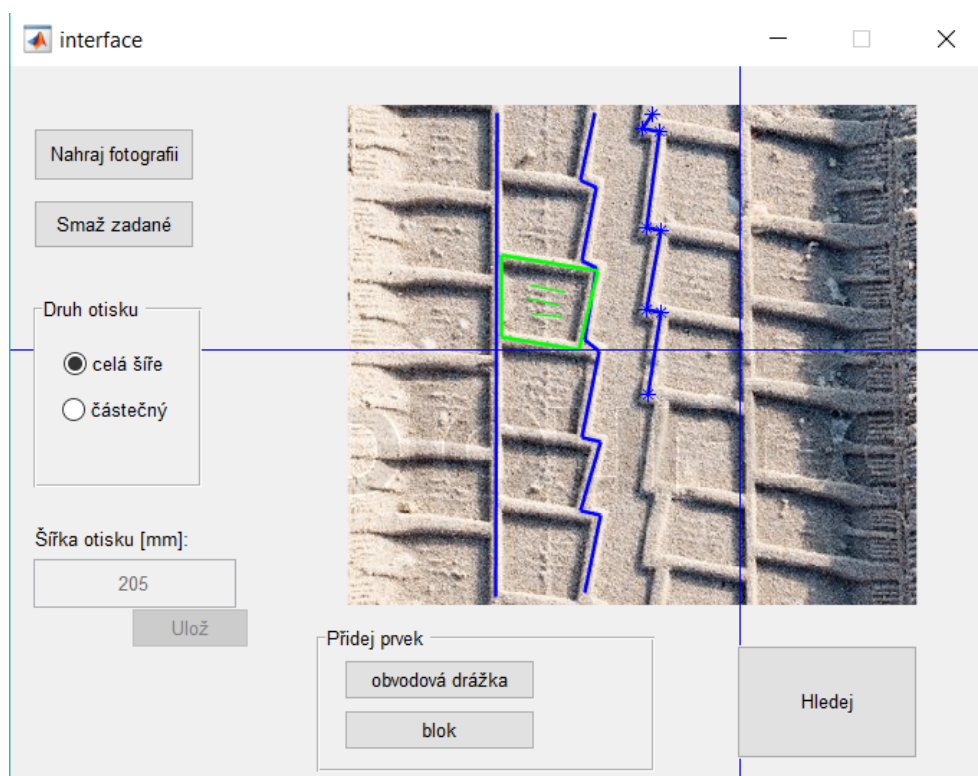
V případě této práce bylo upřednostněno grafické rozhraní (obr. 3.4) na úkor příkazové řádky.



Obrázek 3.4: Grafické rozhraní Application Compiler

Více o dané problematice se můžete dočíst v bakalářské práci „*Tvorba samostatně spustitelných aplikací v MATLABu*“ od Jindřicha Vlasáka[23].

3. REALIZACE



Obrázek 3.5: Výsledný produkt

Testování a vyhodnocení

Tato kapitola se zabývá testováním vyvinutého rozhraní. Zaměřuje se nejenom na správnou funkčnost, ale i na spokojenost uživatelů s rozhraním. Nejprve je diskutována kvalita nalezených výsledků vzhledem k předpokládanému výstupu. Posléze se kapitola zabývá zpětnou vazbou od uživatelů, jejich spokojeností s ovládacími prvky rozhraní se zaměřením na uživatelskou přívětivost.

4.1 Testování

Testování prototypu programu bylo prováděno pomocí několika vzorků, o kterých bylo předem známo, zda se v databázi nacházejí či nikoliv.

Fotografie otisků musejí být nejprve upraveny do požadovaného formátu, kdy šířka otisku se rovná šířce fotografie a otisk je orientován souběžně s vertikální osou. Testovaným subjektům byly předány již předzpracované fotografie, aby se ihned mohli zaměřit na testování rozhraní aplikace.

Před začátkem testování je potřeba, aby si subjekt *nastudoval* jak teorii ke vzorkům pneumatik (kapitola č. 1), tak i pokyny pro obsluhu programu, které jsou uvedené v uživatelské příručce (příloha B). Testování samotného GUI začíná v okamžiku nahrání fotografie.

4.1.1 Správnost výsledků

Nalezené výsledky se mají v hlavních parametrech shodovat, na tuto skutečnost je přihlíženo především. Není tedy možné, aby otisk čistě blokového vzorku byl vyhodnocen jako podobný se vzorkem, který má ve středu pneumatiky žebro.

Dalším důležitým faktorem je šířka pneumatiky. Program by neměl nabízet vzorky, které mají rozdílné rozměry. U částečných otisků by neměl být vyhodnocen záznam jako vhodný kandidát pokud šířka otisku je větší než šířka celé pneumatiky.

Sledována je i chyba způsobená stylem zaznamenávání prvků do fotografie. Např. u vzorků, které mají drážky po okrajích středního pásma¹², může výsledek ovlivnit zda je drážka vyznačená podle vnějšího či vnitřního okraje.

4.1.2 Uživatelská přívětivost

Přívětivost rozhraní je testována pomocí 3 subjektů. Každému je předloženo několik otisků, které má za úkol zpracovat pomocí aplikace. Průběh zadávání je pozorován autorkou a případné problémy jsou zaznamenány. Do testování nesmí být zasahováno nikým jiným než testovaným subjektem.

Po ukončení práce s aplikací je provedena zpětná vazba se subjektem. V rámci této zpětné vazby subjekt zhodnotí přehlednost a jednoduchost rozhraní, také má možnost navrhnout zlepšení existujících nebo implementování nových funkcionalit.

4.2 Vyhodnocení

Ve většině vyhledávání program pracuje podle očekávání a navrhuje vzorky podobné hledanému typu. Přestože pneumatika pro daný otisk není obsažena v databázi, navržené výsledky se podobají hledanému vzorku.

Bylo by možné dosáhnout i přesnějších výsledků přidáním dalších parametrů sledovaných na otisku, ale s takto malou databází to není potřeba.

4.2.1 GUI

Z testování uživatelského prostředí vyplynulo, že je nejprve nutné si na způsob zadávání zvyknout. S každým dalším zpracováním otisku se sebejistota a rychlost subjektu u zadávání zvyšovala.

Z počátku subjekty občas zapoměly potvrdit zadání jednotlivých prvků stisknutím klávesy *Enter*, a byly proto nuceny absolvovat celý postup znovu od začátku. Při pochybení na začátku byla funkce Smaž zadané dostatečná, ale při pochybení v pokročilejší fázi vyznačování bylo navrácení na začátek velice nepříjemné. Je tedy vhodné přidat do rozhraní ovládací prvek, který umožní krokované **navracení zpět** v posloupnosti zadaných prvků.

V jednom případě subjekt chtěl začít zadávat od znova, ale protože se zrovna nacházel v módu zadávání, nedokázal se vrátit jinak, než vypnutím a opětovným zapnutím celé aplikace. Z této situace vzešel dobrý podnět na vytvoření malé **nápovědy/navigace** s informacemi aktuálními pro daný mód (např. stisknutím klávesy *Esc* opustíte zadávání).

¹²oblast podél středu dezénu pneumatiky

Závěr

Program MATLAB se ukázal jako všestranný nástroj schopný splnit všechny potřebné funkcionality pro danou aplikaci počínaje vytvořením GUI až po transformování do samostatně spustitelné aplikace. Aplikaci se podařilo implementovat bez větších problémů za použití pouze MATLABu, bez nutnosti využívání externích programů.

Proběhlé testování přineslo uspokojivé výsledky a prokázalo tak funkčnost aplikace. Největší hrozbou pro aplikaci je neodborné použití, kterému se snaží předcházet kvalitním proškolením uživatele. Moderní technologie automatického zpracování obrazu a jejich vývoj se posouvá čím dál více kupředu. Nasnadě je tedy využití těchto technologií.

Do budoucna by bylo tedy lepší vydat se cestou automatického zpracování obrazu, místo ručního zadávání operátorem. Vyloučila by se tak lidská chyba a zároveň by se i zpracování a vyhledávání značně urychlilo. Zadání této práce však vyžadovalo ruční převod obrazových dat. V používaném prostředí MATLAB je např. k dispozici *Image Processing Toolbox* [24], který by bylo možné pro automatické zpracování obrazu využít.

Literatura

- [1] Zdeněk, J.; Ždánský, B.; Čupera, J.: *Automobily I – Podvozky*. AVID, 2010.
- [2] Sajdl, J.: Diagonální pneumatika [online]. [cit. 2017-10-20]. Dostupné z: <http://www.autolexicon.net/cs/articles/diagonalni-pneumatika/>
- [3] Symetrické, asymetrické, směrové pneumatiky: jak si vybrat? [online]. [cit. 2017-10-20]. Dostupné z: <http://www.rezulteo-pneumatiky.cz/pruvodce-pneumatik/dobre-koupit-pneumatiky/symetricke-asymetricke-smerove-pneumatiky-jak-si-vybrat-23680>
- [4] Bodziak, W. J.: *Tire Tread and Tire Track Evidence: Recovery and Forensic Examination*. CRC Press, 2008, ISBN 9780849372476.
- [5] Liu, Y.; Yan, H.; Lim, K.-P.: Study on rotation-invariant texture feature extraction for tire pattern retrieval. *Multidimensional Systems and Signal Processing*, ročník 28, č. 2, Apr 2017: s. 757–770, ISSN 1573-0824, doi:10.1007/s11045-015-0373-0. Dostupné z: <https://doi.org/10.1007/s11045-015-0373-0>
- [6] Dhawan, A. P.: *Medical Image Analysis*. John Wiley Publications and IEEE Press, 2003.
- [7] Lux, F. H.: Tire Track Identification. *Journal of Forensic Research*, ročník 4, č. 4, 2013, doi:10.4172/2157-7145.1000198. Dostupné z: <https://www.omicsonline.org/tire-track-identification-2157-7145.1000198.php?aid=19229>
- [8] A database of tyre tread patterns for the identification of crime scene marks [online]. [cit. 2017-10-20]. Dostupné z: <http://www.fosterfreeman.com/index.php/component/content/article/20-products/shoe-print-identification/119-treadmate>

- [9] Wang, Y.-W.: A tire-mark identification scheme for suspected vehicle detection in hit and run accident. *Journal of the Eastern Asia Society for Transportation Studies*, 01 2005.
- [10] Number of passenger cars and commercial vehicles in use worldwide from 2006 to 2015 in (1,000 units). 2017, [cit. 2017-12-01]. Dostupné z: <https://www.statista.com/statistics/281134/number-of-vehicles-in-use-worldwide/>
- [11] Konstrukce, funkce a výroba pneumatiky [online]. [cit. 2017-10-20]. Dostupné z: http://www.pneu-asistent.cz/Konstrukce_funkce_a_vyroba_pneumatiky.html
- [12] Valský, T.: Typy dezénu pneumatik [online]. 2013, [cit. 2017-10-20]. Dostupné z: <https://www.pneumatiky.cz/info/typy-dezenu-pneumatik.html>
- [13] Usanov, D. A.; Kalinkin, M. Y.; Skripal', A. V.: Identification of a Tire-Tread Pattern from Its Track Using a Computer-Aided Video Technique. *Russian Journal of Nondestructive Testing*, ročník 41, č. 6, Jun 2005: s. 382–386, ISSN 1608-3385, doi:10.1007/s11181-005-0180-z. Dostupné z: <https://doi.org/10.1007/s11181-005-0180-z>
- [14] Kingsbury, N.: Complex Wavelets for Shift Invariant Analysis and Filtering of Signals. *Applied and Computational Harmonic Analysis* 10, 2001, ISSN 1063-5203, doi:10.1006/archa.2000.0343. Dostupné z: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.588.4232&rep=rep1&type=pdf>
- [15] Cervinka, L.; Cip, P.; Horak, K.: Characters classifications by Radon transform. *Annals of DAAAM & Proceedings*, Jan 2010, ISSN 1726-9679.
- [16] The MathWorks, Inc.: *XML Documents* [online]. [cit. 2017-12-25]. Dostupné z: <https://www.mathworks.com/help/matlab/xml-documents.html>
- [17] Library of Tire Tread Patterns [online]. [cit. 2017-11-13]. Dostupné z: <http://www.biologyjunction.com/tire%20tread%20patterns.html>
- [18] The MathWorks, Inc.: *MATLAB GUI* [online]. [cit. 2017-12-27]. Dostupné z: <https://www.mathworks.com/discovery/matlab-gui.html>
- [19] The MathWorks, Inc.: *GINPUT* [online]. [cit. 2017-12-27]. Dostupné z: <https://www.mathworks.com/help/matlab/ref/ginput.html>
- [20] The MathWorks, Inc.: *Custom GINPUT* [online]. [cit. 2017-12-27]. Dostupné z: <https://www.mathworks.com/matlabcentral/fileexchange/38703-custom-ginput>

- [21] The MathWorks, Inc.: *Application Compiler [online]*. [cit. 2017-12-29]. Dostupné z: <https://www.mathworks.com/help/compiler/applicationcompiler-app.html>
- [22] The MathWorks, Inc.: *applicationCompiler [online]*. [cit. 2017-12-29]. Dostupné z: <https://www.mathworks.com/help/compiler/applicationcompiler.html>
- [23] Vlasák, J.: Tvorba samostatně spustitelných aplikací v MATLABu. 2010, [cit. 2018-01-02]. Dostupné z: <https://dk.upce.cz/handle/10195/37046>
- [24] The MathWorks, Inc.: *Image Processing Toolbox [online]*. [cit. 2018-01-03]. Dostupné z: <https://www.mathworks.com/help/images/>

Seznam použitých zkratek

1D jedno-dimenzionální

2D dvou-dimenzionální

3D tří-dimenzionální

GUI Graphical User Interface

XML Extensible Markup Language

MATLAB MATrix LABoratory

Radon-DTCWT Radon Dual Tree Complex Wavelet Transform

CEDA Curvelet Energy Distribution Algorithm

RT Radonová Transformace

GUIDE Graphical User Interface Development Enviroment

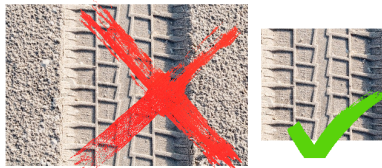
DOM Document Object Model

ASCII American Standard Code for Information Interchange

Uživatelská příručka

Požadavky na vstupní data

1. Šířka fotografie musí přesně odpovídat šířce otisku na fotografii (obr. B.1)



Obrázek B.1: Příklad vstupu (zleva špatně, správně)

2. Otisk musí být orientován kolmo k vodorovné ose - šířka pneumatiky se odečítá z vodorovné osy (obr. B.2)



Obrázek B.2: Příklad vstupu (zleva špatně, správně)

Posloupnost zadávání

Krok 1 - Nahrání fotografie

Pomocí tlačítka *Nahraj fotografii* se otevře průzkumník souborů. Vyhledejte požadovaný soubor, označte ho kliknutím myši a stiskněte tlačítko *Otevřít*.

Krok 2 - Druh otisku

Pomocí nabídky *Druh otisku* zvolte zda se jedná o fotografii celé šířky otisku nebo pouze o otisk částečný. V závislosti na této volbě se vybere algoritmus pro následné hledání.

Krok 3 - Zadání šířky otisku

Do pole *Šířka otisku [mm]* zadejte skutečnou šířku otisku v milimetrech. Zadání hodnoty potvrďte následným stisknutím tlačítka *Ulož*.

Krok 4 - Vyznačování prvků

V sekci *Přidej prvek* vyberte prvek, který chcete v otisku vyznačit. Kurzor myši se přepne do módu záznamu daného prvku. Vyznačit lze přímé i obecné křivky, kde se vše zadává pomocí krajních bodů vytvořených kliknutím myši (pravé či levé tlačítko).

Při stisknutí **levého tlačítka** myši se vykreslují body, které se propojují přímkou. Pokud chcete zakreslit oblouk klikněte levým tlačítkem (začátek oblouku) následně použijte **pravé tlačítko** myši na označení oblouku a opět levé tlačítko (konec oblouku). Při stisknutí pravého tlačítka myši se spojnice nevykresluje, čeká se na bod ukončující oblouk.

Obvodové/Laterální drážky - zadání drážky vždy ukončete stisknutím klávesy *Enter*.

Blok - bloky nezavírejte manuálně, po stisknutí klávesy *Enter* se automaticky první a poslední bod spojí. Následně se dostanete do módu zadávání **lamel** (malých drážek) uvnitř bloku. Tyto lamely se zadávají stejným způsobem, každá je třeba potvrdit klávesou *Enter*. Po vyznačení všech lamel odejdete z tohoto módu stisknutím klávesy *Esc*.

Ve směru šířky otisku je třeba vyznačit všechny prvky postupně od levého k pravému okraji. Takto se získá počet a umístění jednotlivých prvků napříč otiskem pneumatiky.

Krok 5 - Hledání

Po zahájení hledání stisknutím tlačítka *Hledej* se zobrazí nabídka kandidátů na shodu v novém okně. Společně s touto nabídkou je zobrazeno i dialogové okno na zadání čísla pneumatiky, která odpovídá otisku. Po zadání a potvrzení volby (0 pro nenalezeno) se zobrazí příslušný záznam z databáze.

Volitelný krok - Smazání vyznačeného

Stisknutím tlačítka *Smaž zadané* se z fotografie smažou všechny zadané prvky i uložená data z paměti.

Obsah přiloženého CD

| | | |
|--|------------------|---|
| | readme.txt..... | stručný popis obsahu CD |
| | exe | adresář se spustitelnou formou implementace |
| | src | |
| | impl..... | zdrojové kódy implementace |
| | thesis | zdrojová forma práce ve formátu L ^A T _E X |
| | text | |
| | thesis.pdf | text práce ve formátu PDF |
| | thesis.ps | text práce ve formátu PS |