České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra počítačové grafiky a interakce

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: Bc. Martin Janda

Studijní program: Otevřená informatika
Obor: Počítačová grafika a interakce

Název tématu: Vizualizace n-rozměrných heterogenních dat

Pokyny pro vypracování:

Analyzujte metody pro vizualizaci n-rozměrných dat [1]. Na základě analýzy navrhněte a implementujte vizualizaci n-rozměrných heterogenních dat, tedy takových, kde data v rozdílných dimenzích nabývají rozdílných typů (reálná čísla, celá čísla, procenta, 1 z N, binární hodnota). Taková data se běžně vyskytují např. v dotaznících. Umožněte filtrování dat (brushing) a zobrazení více než jednoho filtru pro možnost porovnání dvou či více množin vyfiltrovaných dat. Výslednou aplikaci otestujte alespoň na třech různých vstupních datech o různé složitosti (různý počet dimenzí). Výslednou aplikaci porovnejte pomocí uživatelských testů (subjektivní evaluace) s paralelními množinami [2].

Seznam odborné literatury:

[1] Heinrich, J., & Weiskopf. State of the art of parallel coordinates. STAR Proceedings of Eurographics, 95-116, 2013.
[2] Kosara, R., Bendix, F., & Hauser, H. Parallel sets: Interactive exploration and visual analysis of categorical data. IEEE Transactions on Visualization and Computer Graphics, 12(4), 558-568, 2006.

Vedoucí: Ing. Ladislav Čmolík, Ph.D.

Platnost zadání: do konce zimního semestru 2018/2019

[redacted]
prof. Ing. Jiří Žára, CSc.
vedoucí katedry

L.S.

[redacted]
prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 3.4.2017

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra počítačové grafiky a interakce

Vizualizace n-rozměrných heterogenních dat

Visualization of n-dimensional heterogenous data

Diplomová práce

Studijní program: Otevřená informatika
Studijní obor: Počítačová grafika a interakce

Vedoucí práce: Ing. Ladislav Čmolík, Ph.D.

Bc. Martin Janda

Praha 2018

Poděkování

Chtěl bych poděkovat vedoucímu práce Ing. Ladislavu Čmolíkovi, Ph.D. za odborné vedení a cenné rady. Dále bych chtěl poděkovat účastníkům uživatelských testů, bez kterých by nemohly být výsledky práce ověřeny.

Prohlášení

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

V Praze dne: …….. …….

(podpis autora)

Abstrakt

Tato diplomová práce se věnuje návrhu a implementaci aplikace pro vizualizaci n-rozměrných heterogenních dat. Nejdřív jsou zde analyzovány metody, které se pro vizualizaci n-rozměrných dat používají, a jsou popsány jejich výhody a nevýhody. Poté je popsána nově navržená vizualizační metoda, která vychází z metody paralelních množin, a snaží se odstranit její nedostatky. Další část práce se věnuje popisu návrhu a implementace aplikace, která umožňuje vizualizaci dat pomocí nově navržené metody i klasické metody paralelních množin. V závěru práce je navržená aplikace otestována uživatelskými testy s cílem porovnat dvě použité vizualizační metody.

Klíčová slova: vizualizace dat, n-rozměrná heterogenní data, paralelní množiny, Set Rivers

Abstract

This master thesis is concerned with a design and implementation of application for visualization of n-dimensional heterogeneous data. Firstly, there are analysed methods used for visualization of n-dimension data with a description of their advantages and disadvantages attached. Then the thesis pays attention to a newly designed visualization method. The base of the method is related to parallel sets method and it tries to remove its defects. In another part, there is description of the design and implementation of the application that can visualize data through the newly designed method and method of parallel sets. At the end, one can find the designed application tested by user-test with the purpose to compare the used methods.

Key words: visualization, n-dimensional heterogeneous data, parallel sets, Set Rivers

Obsah

1 Úvod ... 13
2 Analýza ... 15
2.1 Scatter plot ... 15
2.2 Scatter plot matrix ... 18
2.3 Glyphs ... 19
2.4 Paralelní souřadnice ... 21
2.5 Mosaic plot ... 24
2.6 Paralelní množiny ... 25
2.7 Shrnutí analýzy ... 27
3 Návrh řešení ... 29
3.1 Vylepšení metody paralelních množin ... 29
3.1.1 Analýza omezení paralelních množin ... 29
3.1.2 Popis metody Set Rivers ... 31
3.2 Požadavky na systém ... 34
3.3 Navrhovaná struktura aplikace ... 34
3.4 Návrh procesů ... 36
3.4.1 Načtení dat ... 37
3.4.2 Tvorba kategorií ... 37
3.4.3 Řazení kategorií ... 38
3.4.4 Tvorba propojujících rovnoběžníků ... 39
3.4.5 Výpočet výšky a pozice rovnoběžníků ... 40
3.4.6 Generování barev ... 41
3.4.7 Vykreslení grafu ... 42
3.4.8 Návrh základních způsobů interakce ... 43
3.4.9 Editace kategorií ... 45
3.4.10 Vlastní osa ve stromovém rozložení ... 45
3.4.11 Filtry ... 47
3.4.12 Rovnoběžníky uvnitř os ... 49
3.4.13 Rozšíření osy o souřadnice ... 50
4 Implementace ... 53
4.1 Použité technologie ... 53
4.1.1 XDat ... 53

4.2 Struktura tříd 54
5 Výsledky 57
5.1 Vstupní data 57
5.2 Metoda paralelních množin 57
5.2.1 Stromové rozložení 58
5.2.2 Svazkové rozložení 60
5.2.3 Vlastní osa 61
5.3 Metoda Set Rivers 61
5.3.1 Jeden filtr 63
5.3.2 Více filtrů 64
5.3.3 Číselná osa 65
6 Uživatelské testování 67
6.1 Popis testovací metody a metod pro zpracování výsledků 67
6.2 Průběh tesu 68
6.3 Testovací data 69
6.4 Testovací úkoly 71
6.5 Nálezy 72
6.5.1 Metoda 1 – paralelní množiny 72
6.5.2 Metoda 2 – Set Rivers 73
6.6 Statistické zpracování výsledků 73
6.7 Zhodnocení výsledků testu 75
7 Závěr 77
8 Literatura 79

Seznam příloh

A Úkoly pro uživatelský test 83
B Dotazník pro uživatelský test 85

1 Úvod

V reálném světě se často vyskytují data tabulárního charakteru, která mají velký počet atributů (sloupců tabulky). Jedná se například o data z dotazníků, ze senzorů měřících zařízení nebo o data popisující parametry různých objektů nebo služeb. Taková data mohou nabývat ve všech parametrech spojitých číselných hodnot, a potom patří do kategorie spojitých n-rozměrných dat. Pro vizualizaci tohoto typu dat existují vizualizační metody jako scatter plot matrix [6] nebo paralelní souřadnice [7], které dosahují dobrých výsledků a nabízejí přehlednou vizualizaci pro možnost analýzy dat. Ve většině reálných případů se ale jedná o data, která nabývají kategorických hodnot nebo mají dokonce některé atributy kategorické a jiné spojité. To platí například právě pro data z dotazníků, kde jedna odpověď může být číselného typu, jiná výběr z několika možností a další například jen ANO / NE. Existují i vizualizační metody, které se zaměřují na vizualizaci kategorických n-rozměrných dat jako například mosaic plot [8] nebo paralelní množiny [12], ale mají své omezení a nehodí se pro všechny případy dat a vizualizačních úloh. Ještě větší problém nastává v případě již zmíněných dat, kde atributy nabývají různých typů. Taková data se nazývají heterogenní a možnosti pro jejich vizualizaci jsou poměrně omezené.

Heterogenní n-rozměrná data tedy není jednoduché vizualizovat, ale přitom právě těchto dat se v reálném světě vyskytuje mnoho a je potřeba s nimi pracovat a analyzovat je. Tento fakt slouží jako motivace pro výzkum v oblasti vizualizace dat. Ing. Ladislav Čmolík Ph.D. navrhl novou vizualizační metodu, která se zakládá na metodě paralelních množin, ale snaží se odstranit některé její nedostatky a poskytnout lepší výsledky právě pro heterogenní data. Novou metodu je ale třeba prakticky ověřit na reálných datech, a proto si tato práce klade za cíl navrhnout a implementovat aplikaci, která umožní vizualizaci dat pomocí této nové metody. Pro možnost srovnání bude aplikace umožnovat i vizualizaci pomocí tradiční metody paralelních množin.

V této práci jsou nejdřív rozebrány metody, které by mohly být pro vizualizaci heterogenních n-rozměrných dat nejvhodnější. Poté jsou analyzovány jejich problémy a je popsáno, jak se je nová metoda snaží řešit. Následuje návrh aplikace, kterému se věnuje největší část této práce. Jsou zde popsány procesy i použité algoritmy pro řešení problémů. Dále byl jako součást této práce proveden uživatelský test, jehož hlavním účelem je právě srovnání dvou vizualizačních metod a ověření, jestli nově navržená metoda skutečně řeší problémy metody paralelních množin. Aby bylo možné pomocí uživatelského testu metody srovnat, byly pozvány dvě skupiny uživatelů, kde jedna řešila připravené úkoly v navržené aplikaci pomocí metody paralelních množin a druhá řešila stejné úkoly pomocí nově navržené metody. Kvůli eliminaci možných rozdílů v počátečních zkušenostech uživatelů s vizualizací dat byla před samotný test zařazena tréninková fáze, kde se

uživatelé seznámili s aplikací i vizualizační metodou. Pro test a tréninkovou fázi byly použity dva různé soubory reálných dat, které se od sebe lišily, ale měly stejný charakter. Tento test tedy nabízí první pohled na použití nové metody v praxi a porovnání s běžně používanou metodou paralelních množin. Výsledky testu jsou reportovány v závěrečné části této práce.

2 Analýza

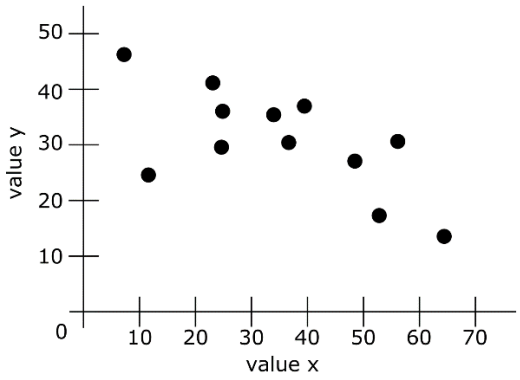
N-rozměrná data jsou data tabulárního charakteru, kde řádky tabulky reprezentují jednotlivé položky dat a sloupce reprezentují atributy (dimenze dat). Buňky tabulky potom obsahují skalární hodnoty, kterých nabývá daná položka dat pro daný atribut [26]. Příkladem takových dat jsou konfigurace a parametry různých zařízení, nebo například data z průzkumů trhu a dotazníků. Charakter takových dat může být buďto spojitý, potom všechny atributy mají spojitý charakter a jejich hodnoty jsou reálná čísla, nebo kategorický, kde hodnoty pro všechny atributy spadají do diskrétních kategorií. Nebo se může jednat o data, která mají některé atributy spojitého charakteru a některé kategorické.

Při analýze n-rozměrných dat nás zajímají různé typy dotazů, které můžeme rozdělit do tří skupin [26]. První skupina jsou dotazy na jednotlivé položky, což může být například dotaz, jakých hodnot nabývá položka pro dané atributy. Další skupina jsou dotazy na atributy, jako například jaké je rozložení jednotlivých položek dat pro daný atribut. Typicky nejzajímavější je poslední skupina dotazů, kde se ptáme na relace mezi jednotlivými atributy. Můžeme vyšetřovat existenci a typ korelace mezi atributy nebo například analyzovat shluky dat (clustering).

Se zodpovězením těchto dotazů nám pomáhají vizualizační techniky, které jsou určeny pro n-rozměrná data. Příklady takových technik jsou: scatter plot matrix [6], glyphs [27], paralelní souřadnice [7] a paralelní množiny [12]. Každá z uvedených technik je vhodná pro jiné použití a jiný charakter dat. Popisu těchto technik a jejich použití je věnována tato kapitola.

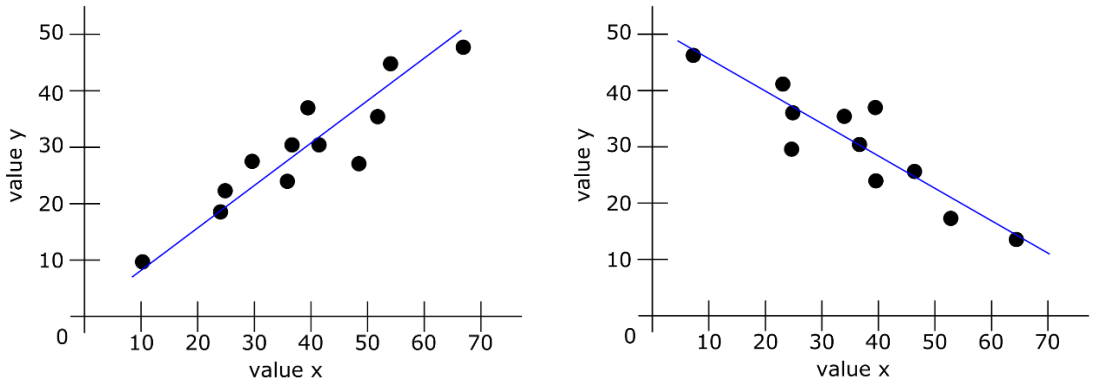
2.1 Scatter plot

Scatter plot [25], neboli bodový graf je technika, kde se hodnoty dvou proměnných zobrazují do kartézského souřadného systému. Data jsou zobrazena jako množina bodů, kde hodnota první proměnné udává pozici bodu na svislé ose a hodnota druhé proměnné udává pozici na vodorovné ose (Obrázek 1).



Obrázek 1: Scatter plot

Toto zobrazení umožňuje jednoduchou identifikaci závislostí dvou proměnných mezi sebou. Při vizualizaci dvou spojitých proměnných x a y lze z grafu jasně vyčíst, jestli proměnná y je závislá na x, případně jestli závislost je lineární, kvadratická nebo jiná. V případě existence závislosti mezi proměnnými lze body proložit přímku nebo křivku a závislost dále popsat (Obrázek 2). Také lze jednoduše identifikovat extrémní hodnoty, které se odlišují od ostatních vzorků.

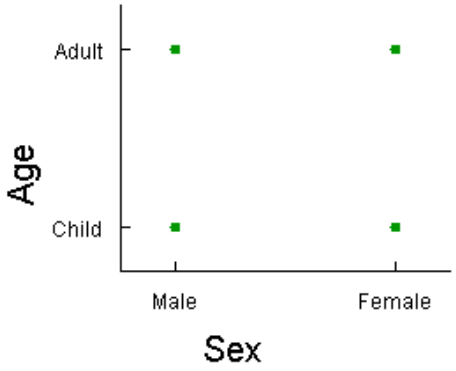


(a) *(b)*

Obrázek 2: Závislost x, y – (a) přímka z levého dolního rohu do pravého horního rohu – pozitivní závislost, (b) přímka z levého horního rohu do pravého dolního rohu – negativní závislost

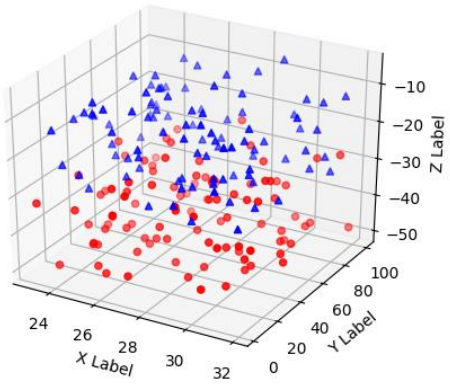
Tato vizualizační metoda také umožňuje metodu interakce zvanou brushing [2]. Jedná se o zobrazení nebo zvýraznění jen určité části dat, která spadá do oblasti zájmu uživatele. Interakce probíhá tak, že uživatel vybere intervaly hodnot nezávisle pro každou proměnnou a následně se v grafu zvýrazní pouze ta data, která pro obě proměnné spadají do zadaných intervalů. Zvýraznění je realizováno tak, že se vybraná data obarví jinou barvou než ostatní nebo se ostatní data úplně skryjí. Pomocí toho lze dosáhnout přehlednějšího zobrazení požadovaných dat.

Předpokladem pro zobrazení technikou scatter plot je ale podmínka, že obě proměnné nabývají spojitých hodnot. V případě, že se jedná o kategorické hodnoty, všechna data leží v několika málo bodech a informace o jejich množství a závislosti proměnných se ztratí (Obrázek 3).



Obrázek 3: Zobrazení kategorických dat technikou scatter plot – atributy věk a pohlaví z reálných dat cestujících na Titaniku

Další omezení základní metody scatter plot je, jak již bylo zmíněno, že metoda zobrazuje jen hodnoty dvou proměnných. To by v aplikaci na vícerozměrná data znamenalo možnost zobrazit pouze data dvourozměrná. Existuje několik způsobů, jejichž snahou je počet proměnných zvýšit. Jeden z nich je 3D scatter plot [13], který rozšiřuje dvourozměrný graf do 3D a tím přidává možnost zobrazit třetí proměnnou (Obrázek 4). Identifikace pozice bodů ve 3D je ale mnohem obtížnější a ve statickém obrázku na 2D monitoru ani nemusí být možná. Proto je zavedena ještě interakce rotace, díky které lze pozici bodů identifikovat. Identifikace je ale i tak poměrně obtížná a celý graf je mnohem méně přehledný než klasický scatter plot ve 2D. Další možností přidání další proměnné je zakódování jejích hodnot do barvy nebo tvaru bodů. Takto je ale možné zobrazit jen malé množství různých hodnot a sledování závislostí mezi proměnnými je obtížné.

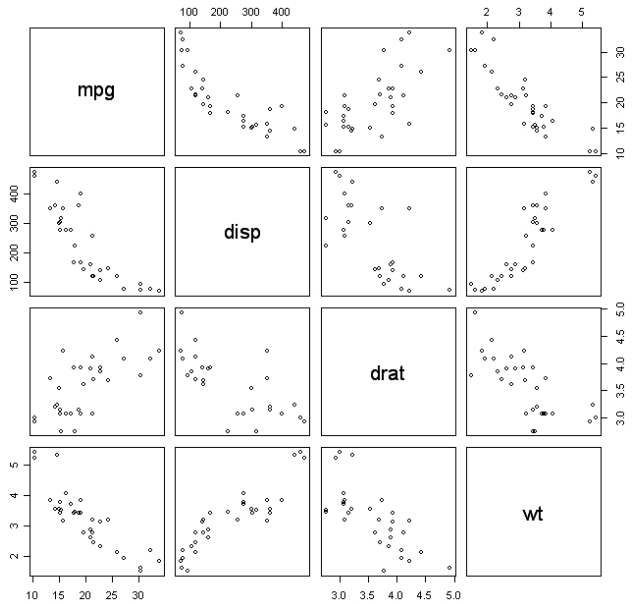


Obrázek 4: 3D scatter plot [16]

I při použití metod rozšiřující scatter plot o další proměnné, je ale stále počet proměnných velmi limitovaný, a proto použití této metody pro obecná n-rozměrná data není možné.

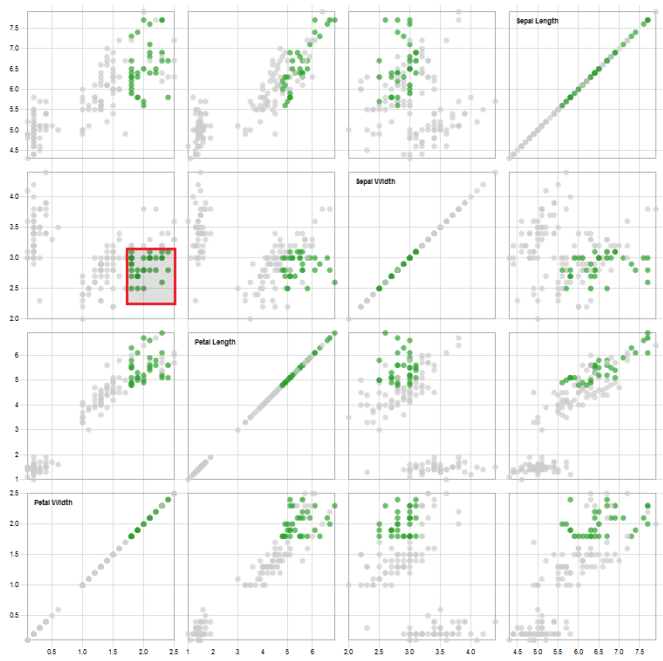
2.2 Scatter plot matrix

Jak bylo uvedeno výše, technika scatter plot není vhodná pro n-rozměrná data. Přesto se zmíněná technika pro tato data použít dá v podobě scatter plot matrix [6]. Vizualizační technika scatter plot matrix využívá scatter plot pro vizualizaci všech dvojic atributů mezi sebou a výsledné vizualizace zobrazí najednou na obrazovce rozdělené do matice. V této matici jsou tedy potom vidět vizualizace všech kombinací dvou atributů mezi sebou (Obrázek 5).



Obrázek 5: Scatter plot matrix [20]

Díky tomu, že tato technika využívá scatter plot, závislosti mezi jednotlivými atributy jsou jednoduše identifikovatelné a je možno použít interaktivní techniku brushing stejným způsobem jako u scatter plot [2]. Při použití techniky brushing uživatel vybere intervaly hodnot pro jednu dvojici atributů, což reprezentuje obdélník uvnitř jednoho ze scatter plot grafů. Data, která spadají do zadaného intervalu se poté zvýrazní ve všech scatter plot grafech (Obrázek 6).



Obrázek 6: Scatter plot matrix brushing [21]

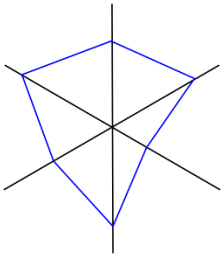
Pomocí této techniky lze tedy jednoduše identifikovat závislosti mezi každými dvěma atributy, ale závislosti mezi více než dvěma atributy zároveň se identifikují velmi obtížně. Další nevýhoda této techniky je, že prostor obrazovky je rozdělen mezi všechny kombinace atributů, takže pro velký počet atributů je tato technika nepoužitelná. Důležitá vlastnost plyne i z použití scatter plot grafů, díky kterým je tato metoda vhodná pouze pro spojitá data. V případě kategorických dat, by v grafech vzniklo jen několik bodů, tak, jak bylo demonstrováno v kapitole 2.1, a vizualizace by byla nepoužitelná.

2.3 Glyphs

Vizualizační technika glyphs mapuje atributy dat na grafické elementy ikon a symbolů [27]. Příkladem takových ikon mohou být šipky, hvězdice s různě dlouhými paprsky nebo i složité tvary jako například ikony tváře. Mapování probíhá tak, že každé položce dat odpovídá jedna grafická ikona a atributy dat reprezentují jednotlivé grafické elementy této ikony. Podle hodnot atributu se poté mění například velikost, barva, tvar nebo orientace daného elementu a závislosti mezi daty je možné sledovat podle podobnosti jednotlivých ikon.

Příklady této vizualizační techniky jsou star glyphs [22] nebo Chernoff faces, které představil Herman Chernoff v roce 1973 [10].

V technice star glyphs jsou jednotlivé položky dat reprezentovány mnohoúhelníky, které tvoří uzavřená lomená čára. Ze středu těchto symbolů jsou vedeny osy, které reprezentují jednotlivé atributy dat. Mapování atributů na grafické elementy je realizováno tak, že lomená čára tvořící mnohoúhelník, protíná každou osu v místě, které odpovídá hodnotě datového vzorku pro daný atribut. Tímto způsobem vznikne hvězdicový tvar, kde délka každého paprsku odpovídá hodnotě daného atributu (Obrázek 7).



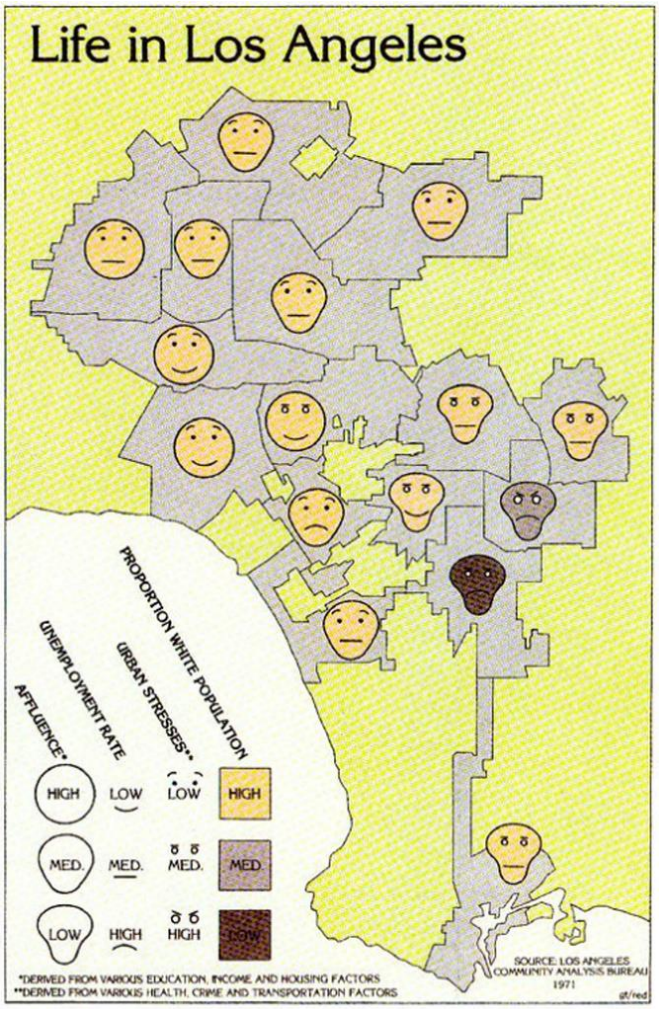
Obrázek 7: Star glyphs

Pro některé atributy platí, že čím menší hodnota, tím lépe a pro jiné naopak (př. spotřeba benzínu vs. maximální rychlost automobilu). Proto je vhodné zavést mapování atributů tak, aby delší paprsek odpovídal „lepší“ hodnotě atributu.

Nevýhoda této techniky je, že pro větší počet atributů je mezi paprsky méně místa, je složitější je oddělit a vizualizace se stává méně přehlednou. Pro dobrou čitelnost by mělo platit, že mezi paprsky bude úhel minimálně 30° [26].

Technika Chernoff faces místo mnohoúhelníků používá pro reprezentaci položek dat obrázky lidské tváře. Atributy jsou potom mapovány na charakteristiky tváře, jako například tvar úst, sklon obočí, velikost a umístění očí nebo tvar lebky. Tato technika využívá přirozené schopnosti člověka snadno a přesně identifikovat výraz lidské tváře [15]. Je vhodné mapovat „dobré“ hodnoty atributů na pozitivní výraz tváře (například šťastný výraz) a „špatné“ hodnoty na negativní (smutný nebo rozzlobený výraz).

Příklad použití techniky chernoff faces je mapa populace Life in Los Angeles, kterou vytvořil Eugene Turner v roce 1977 [19], [23] (Obrázek 8). V tomto případě jsou položky dat jednotlivé geografické oblasti a atributy jsou bohatství, nezaměstnanost, napětí ve městě a podíl bílé populace pro danou oblast. Každý atribut nabývá jen tří hodnot, kterými jsou nízká, střední a vysoká úroveň.



Obrázek 8: Eugene Turner, Life in Los Angeles [23]

Nevýhodou této techniky je jednak omezený počet atributů, které lze vhodně reprezentovat, a zároveň počet hodnot, které atributy nabývají, musí být dostatečně malý, aby byla změna na tváři rozeznatelná. Také sledování závislostí mezi atributy není pomocí této techniky dobře čitelné.

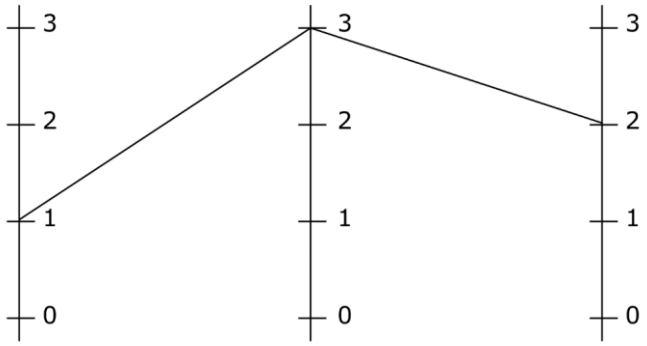
Pro všechny vizualizační techniky glyphs platí také zásadní omezení, kterým je použitelnost pouze pro data s malým počtem položek. Toto omezení vyplývá z principu, že každá položka dat je reprezentována jedním grafickým symbolem.

2.4 Paralelní souřadnice

Další vizualizační technika, určená pro n-rozměrná data, jsou paralelní souřadnice. První zmínka o této technice se objevila ve vědecké literatuře v kontextu Nomogramů [5] a od té doby se paralelní souřadnice staly známou a široce rozšířenou vizualizační technikou pro analýzu dat [7]. Na jejich principu je založeno i mnoho dalších vizualizačních technik jako například N M plot [4] nebo Andrews plot [1].

Charakteristickým znakem pro paralelní souřadnice je paralelní rozmístění os. Každé dimenzi daného prostoru dat odpovídá jedna nezávislá osa, přičemž všechny osy jsou rozmístěny paralelně vedle sebe nebo nad sebou. Orientace os je libovolná, většinou se volí horizontální (osy rovnoběžné s osou x) nebo vertikální (osy rovnoběžné s osou y) rozložení [7]. Volba vhodné orientace závisí na počtu os, rozsahu dat, rozměrech obrazovky i osobních preferencích. Osy jsou obvykle vykresleny jako rovné plné čáry a popisky os jsou umístěny nad nebo pod osou.

Pro N-dimenzionální prostor vypadá graf paralelních souřadnic s vertikální orientací os tak, že je vykresleno *N* os vedle sebe, rovnoběžných s osou y, které tvoří *N*-1 segmentů mezi nimi. Bod, který leží v N-dimenzionálním prostoru, je do paralelních souřadnic zobrazen jako přímky, které protínají osy daných dimenzí v odpovídajících souřadnicích. Každá přímka reprezentuje projekci bodu do roviny určené osami ohraničujícími segment. Vykreslena je pouze část přímek, která leží v segmentu mezi osami. Výsledkem je lomená čára, která protíná všechny osy. Na obrázku (Obrázek 9) je příklad zobrazení bodu *X* o souřadnicích (1,3,2) do paralelních souřadnic.



Obrázek 9: Bod X = (1, 3, 2) v paralelních souřadnicích

Důležitou roli pro zlepšení orientace v datech a vizuální analýzu dat pomocí paralelních souřadnic hraje interakce [7]. Umožňuje uživateli interaktivně měnit parametry podle aktuální potřeby a dostat okamžitou odezvu od systému. Interakci s paralelními souřadnicemi můžeme rozdělit na interakci se vzorky a s osami.

Interakce se vzorky – Brushing

Brushing je běžná interakční technika zavedená původně pro maskování a izolaci bodů ve scatter plot. Tato operace umožňuje uživateli vybrat podmnožinu vzorků, která je použita pro další operace jako zvýrazňování, popisování nebo odstranění. Původně to byla množina, která spadá do osově zarovnaného obdélníku výběru v bodovém grafu. Tento koncept je aplikovatelný i na paralelní souřadnice. Vybrání intervalu na jedné ose paralelních souřadnic odpovídá intervalu v odpovídající dimenzi v prostoru dat. Vybráním intervalů na více osách a jejich skládáním logickými operátory lze omezit prostor dat ve více dimenzích.

Interakce s osami – Translace

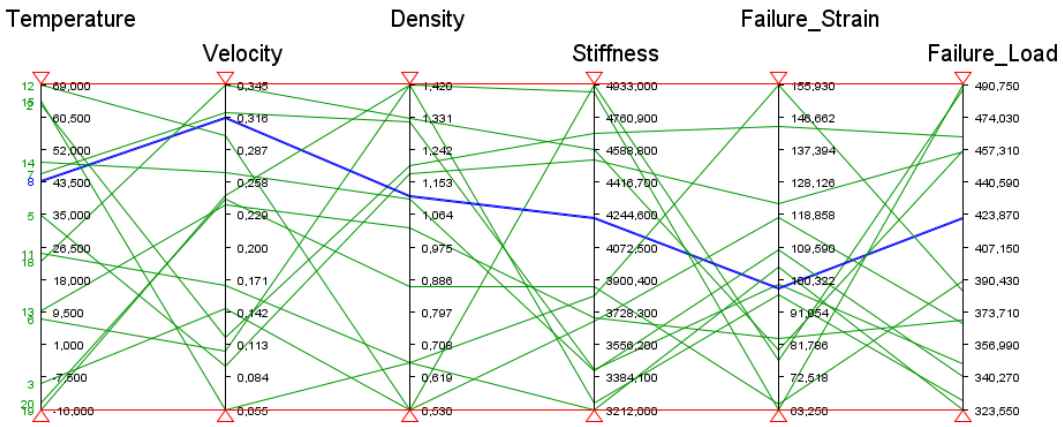
Horizontální pozice os paralelních souřadnic je volný parametr, který lze libovolně měnit. Obvykle se volí vzdálenost mezi všemi osami stejná, někdy ale může být výhodné například zvětšit vzdálenost mezi vybraným párem os kvůli detailnějšímu zobrazení závislostí těchto dvou

dimenzí. Pomocí translace lze také měnit pořadí os a tím zobrazit vzájemnou závislost jiných dimenzí.

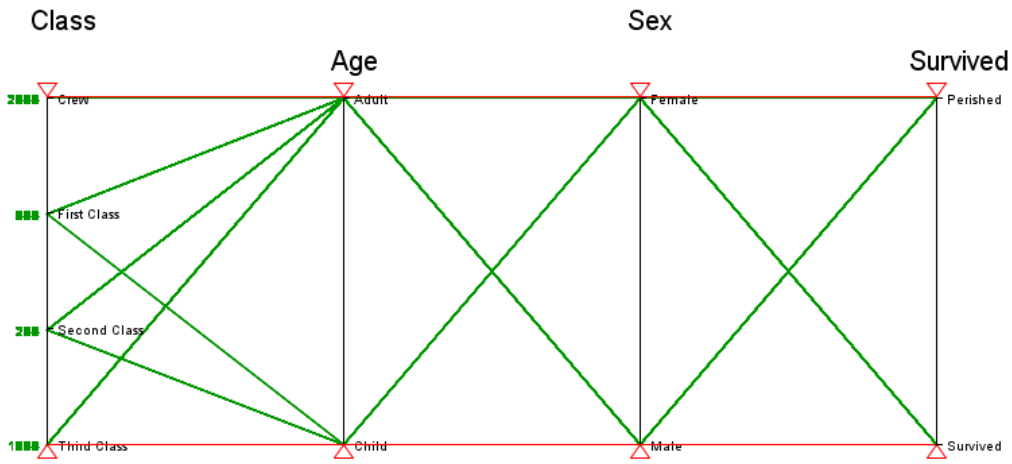
Interakce s osami – Změna měřítka

Základní nastavení měřítka os je takové, že maximální hodnota osy odpovídá maximální hodnotě dat v dané dimenzi a minimální hodnota odpovídá minimální hodnotě dat. To vede k rovnoměrnému rozmístění průsečíků reprezentujících datové vzorky na všech osách, ale každá osa má jiné měřítko. Pro srovnání hodnot mezi osami může být vhodnější zvolit jednotné měřítko všech os.

Výhodou techniky paralelních souřadnic oproti jiným vizualizačním technikám je možnost přehledně zobrazit i větší množství atributů zároveň. Položky dat jsou reprezentovány lomenými čarami, takže vizualizace zůstává přehledná pro mnohem větší počet položek než například u techniky glyphs. Pro extrémně velký počet položek se ale opět přehlednost a čitelnost zhoršuje. Dále tato technika umožňuje přehledně sledovat vzájemné závislosti mezi atributy, a to i mezi více než dvěma zároveň. Nevýhodou ovšem je, že tato technika je opět vhodná jen pro spojité hodnoty atributů. V případě spojitých hodnot se lomené čáry na osách rozprostřou do jednotlivých hodnot na ose a lomenou čáru jsme schopní sledovat napříč grafem (Obrázek 10). V případě kategorických hodnot však budou všechny čáry směřovat do několika málo bodů na ose a nebude tak možné sledovat cestu jednotlivých čar, a tím ani závislosti mezi atributy (Obrázek 11).



Obrázek 10: Paralelní souřadnice – spojité hodnoty



Obrázek 11: Paralelní souřadnice – kategorické hodnoty

2.5 Mosaic plot

Výše popsané techniky včetně paralelních souřadnic fungují dobře pro vizualizaci spojitých n-rozměrných dat, ale pro data, která obsahují atributy s kategorickými hodnotami, nejsou vhodné. Dále v této kapitole jsou představeny vizualizační techniky, které jsou určeny právě pro tento typ dat.

Jedna z uznávaných technik pro vizualizaci kategorických dat je mosaic plot [8]. Tato technika je založena na dělení prostoru do vnořených obdélníků podle atributů dat. Každý obdélník potom reprezentuje určitou kategorii dat a jeho plocha odpovídá počtu datových vzorků, které do této kategorie spadají.

Konstrukce mosaic plot probíhá tak, že se nejdříve určí pořadí atributů dat a každý atribut se přiřadí vertikální nebo horizontální ose. Pro atributy *X1*, *X2*, *X3* by přiřazení vypadalo například následovně: *X1* – horizontální osa, *X2* – vertikální osa, *X3* – horizontální osa. Poté se obdélník, který reprezentuje všechna data, rozdělí v horizontální ose na počet dílů, který odpovídá počtu kategorií atributu *X1* (Obrázek 12a). Plocha těchto nově vzniklých obdélníků odpovídá množství dat v jednotlivých kategoriích atributu *X1*. Dále se každý z těchto obdélníků rozdělí ve vertikální ose podle kategorií atributu *X2* (Obrázek 12b). Vzniknou tedy obdélníky s plochou, která odpovídá množství dat ve všech kombinacích kategorií atributů *X1* a *X2*. Dále se opět každý nový obdélník dělí podle atributu *X3* v ose, která mu byla přiřazena (Obrázek 12c), a takto proces pokračuje přes všechny atributy.

(a) (b) (c)

Obrázek 12: Konstrukce mosaic plot

Pomocí této techniky je tedy možné vizualizovat kategorická data, která mají dva a více atributů. Maximální počet atributů teoreticky není omezen, ale pro více atributů se vizualizace stává nepřehledná. Nevýhoda této techniky je, že v některých případech může být složité porovnat poměry jednotlivých obdélníků a závislosti mezi atributy nejsou dobře viditelné.

2.6 Paralelní množiny

Vizualizační technika paralelní množiny byla představena autory Kosara, Bendix a Hauser [12] a jedná se o vylepšení techniky paralelních souřadnic tak, aby byla vhodná pro kategorická data.

Paralelní množiny kombinují rozložení zobrazení paralelních souřadnic, jako zobrazení nezávislých os pro jednotlivé dimenze vedle sebe, se zobrazením množství dat v diskrétních kategoriích. Spojité osy z paralelních souřadnic jsou rozděleny na oddíly, které reprezentují kategorie, a velikost každého oddílu závisí na množství dat v dané kategorii. Kategorie jsou propojeny rovnoběžníky, které odpovídají množině dat, která patří do obou sousedních kategorií. Tato propojení mají opět šířku závislou na množství dat patřících do dané množiny. Pomocí paralelních množin lze vizualizovat i spojitá data tak, že se spojité hodnoty rozdělí do několika diskrétních kategorií. Vizualizace tedy není omezena počtem položek dat, jako je tomu například u paralelních souřadnic, ale počtem kategorií.

U paralelních množin se dále pro lepší přehlednost používá rozlišení propojujících rovnoběžníků barvami. Jedna dimenze je vybrána jako aktivní (většinou první osa) a definuje barevné kódování pro rovnoběžníky. Každá kategorie aktivní osy má svoji barvu, kterou se obarví i všechny rovnoběžníky, které z ní vycházejí.

Stejně jako u paralelních souřadnic je i u paralelních množin pro detailnější analýzu dat potřeba interakce. Bendix a kolektiv [3] zavedli několik způsobů interakce:

Výběr požadovaných parametrů

Uživatel si zvolí pouze některé dimenze, které se zobrazí ve vizualizaci.

Změna pořadí dimenzí a kategorií

Lze libovolně měnit pořadí os reprezentujících jednotlivé dimenze i pořadí kategorií na jednotlivých osách s okamžitou vizuální odezvou. Změna pořadí os je užitečná pro důkladnější

pohled na závislosti mezi odlišnými dimenzemi. Možnost uživatelsky měnit pořadí kategorií je také vhodná, protože u diskrétních kategorií nemusí existovat žádné přirozené řazení.

Seskupování kategorií

Uživatel může organizovat hierarchii kategorií seskupováním vybraných kategorií.

Skrytí kategorie

Uživatel může odebrat nezajímavé kategorie ze zobrazení a tím efektivněji využít prostor pro zbývající kategorie.

Skládání dimenzí

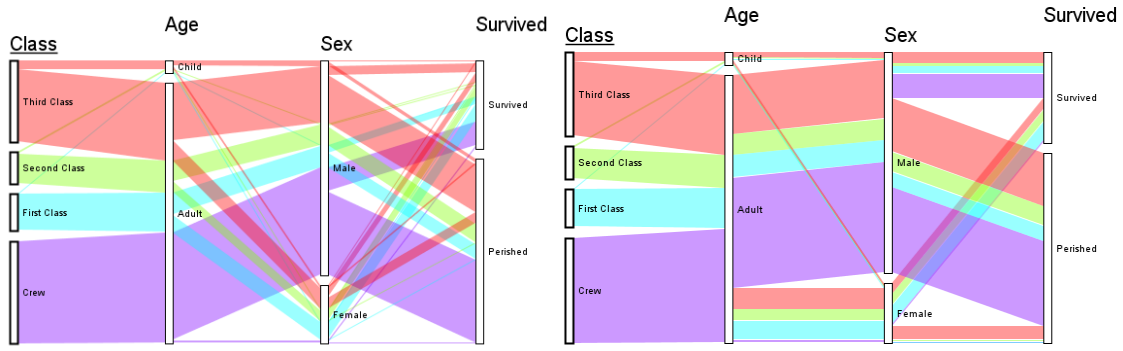
Skládání dimenzí umožňuje uživateli vytvořit novou dimenzi s kategoriemi založenými na parametrech z různých dimenzí. Tím uživatel získá v jedné dimenzi novou klasifikaci dat podle několika dimenzí a počet zobrazených dimenzí ve vizualizaci se může snížit.

Histogram

Uživatel si může zobrazit histogram se statistickými parametry pro detailní analýzu závislostí.

Zvýrazňování

Při najetí kurzoru myši na kategorii se barevně zvýrazní všechny rovnoběžníky, které procházejí danou kategorií.

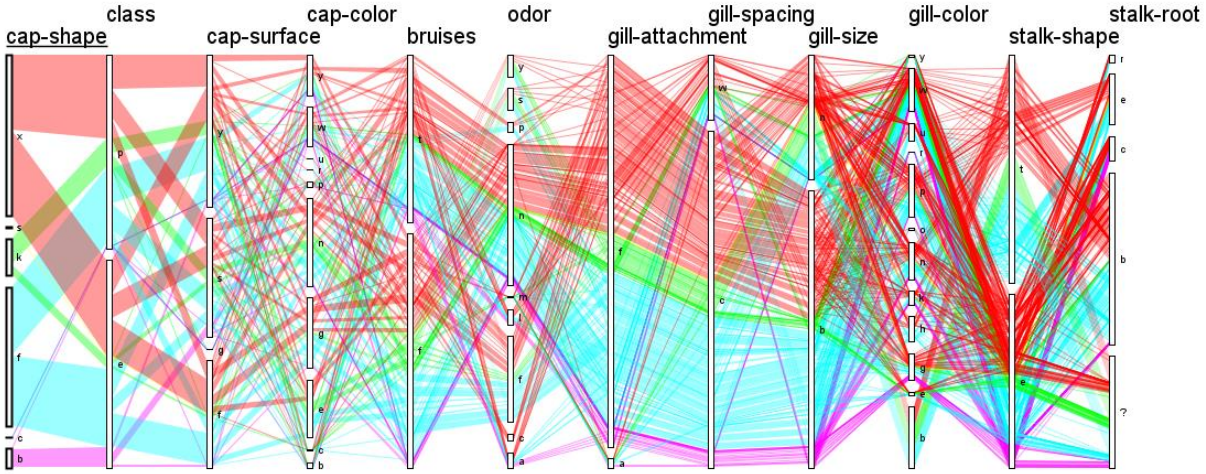


(a) *(b)*

Obrázek 13: Paralelní množiny – stromové rozložení (a), svazkové rozložení (b)

Další optimalizací paralelních množin může být alternativní režim vykreslování propojujících rovnoběžníků mezi kategoriemi. Kosara a kolektiv popsali dva režimy: stromové rozložení (tree layout) a svazkové rozložení (bundled layout) [12]. Při použití stromového rozložení (Obrázek 13a) se rovnoběžníky mezi osami generují od první osy vlevo, což je aktivní kořenová osa, postupně k dalším osám doprava. Na každé ose se rovnoběžníky, které přišly z předchozích os vlevo, dělí na rovnoběžníky, které pokračují do další osy vpravo. Tím vzniká stromová struktura propojujících rovnoběžníků. Tento způsob dělení ovšem vede k tomu, že směrem doprava se mezi osami zvětšuje počet rovnoběžníků, a tím se snižuje přehlednost grafu. (Obrázek 14) Přehlednost se snižuje už při malém počtu os, jako jsou například čtyři, pro větší počet os již přestává být možné z pravé části grafu vyčíst závislosti mezi atributy. Z tohoto důvodu autoři

zavedli ještě svazkové rozložení (Obrázek 13b). Při svazkovém rozložení jsou rovnoběžníky mezi páry sousedních os nezávislé. Rovnoběžník mezi osami je vytvořen pro každou dvojici kategorií z levé a pravé osy v páru, do které patří nějaká data. Obarvení rovnoběžníků se řídí aktivní osou. Pokud by měl jeden rovnoběžník mít víc barev, což znamená, že jeho data spadají do více kategorií v aktivní ose, je tento rovnoběžník rozdělen na barevné pruhy, podle odpovídajících kategorií. Tento způsob rozložení redukuje počet rovnoběžníků a tím dělá graf přehlednější. Je to ovšem za cenu horšího sledování závislostí mezi více osami.



Obrázek 14: Paralelní množiny – nepřehlednost stromového rozložení

2.7 Shrnutí analýzy

Z analýzy vyplývá, že většina vizualizačních technik pro n-rozměrná data je vhodná pouze pro spojité hodnoty atributů. Metod vhodných pro vizualizaci kategorických dat není mnoho a mají své limity a omezení. Například výše popsaná metoda mosaic plot je omezená jednak počtem atributů, pro které je ještě vizualizace přehledná, jednak neumožňuje snadné sledování závislostí mezi atributy. Situace je ještě horší, pokud se jedná o vizualizaci dat heterogenních, které nabývají v některých atributech spojitých hodnot a v jiných kategorických. Jako nejvhodnější metoda pro vizualizaci heterogenních n-rozměrných dat se na základě předchozí analýzy jeví technika paralelních množin.

Paralelní množiny umožnují vizualizovat data se spojitými i kategorickými atributy, umožňují sledování závislostí a počet atributů pro přehlednou vizualizaci není tolik omezen jako u mosaic plot. Přesto má i tato metoda své nevýhody a vznikají zde určité problémy. Při použití stromového rozložení se vizualizace s přibývajícími atributy stává poměrně rychle nepřehlednou a svazkové rozložení zase neumožňuje sledování všech závislostí. Navíc v případě spojitých hodnot atributů je třeba tyto hodnoty rozdělit do kategorií a pracovat těmito atributy jako s kategorickými. Tím pádem nelze sledovat přesné hodnoty datových vzorků.

Na základě analýzy se tedy nedá říci, že by některá z vizualizačních metod pracovala s n-rozměrnými heterogenními daty bez problému a hodila se pro všechny s nimi spojené vizualizační úlohy. V reálném světě se ale právě taková data běžně vyskytují, a proto se jimi tato

práce zabývá. Vedoucí této práce Ing. Ladislav Čmolík, Ph.D. navrhl novou vizualizační techniku určenou pro n-rozměrná heterogenní data, která se snaží uvedené problémy řešit. Popis této techniky je uveden v následující kapitole, dále se práce věnuje návrhu a implementaci aplikace, která kromě vizualizace dat technikou paralelních množin umožňuje také použití nově navržené techniky.

3 Návrh řešení

V této kapitole je nejdřív popsána technika, která se snaží zlepšit metodu paralelních množin a odstranit její nedostatky. Dále se tato kapitola věnuje návrhu aplikace, která kromě vizualizace dat technikou paralelních množin umožňuje také použití nově navržené techniky. Jsou zde definovány požadavky na navrhovaný systém a poté je popsán návrh samotné aplikace.

3.1 Vylepšení metody paralelních množin

Nově navržená technika pro vizualizaci heterogenních n-rozměrných dat, kterou Ing. Ladislav Čmolík, Ph.D. nazval Set Rivers, je založena na technice paralelních množin, ale snaží se odstranit její nedostatky. V této podkapitole nejdřív následuje analýza omezení techniky paralelních množin a dále je popsáno, jak tato omezení metoda Set Rivers řeší.

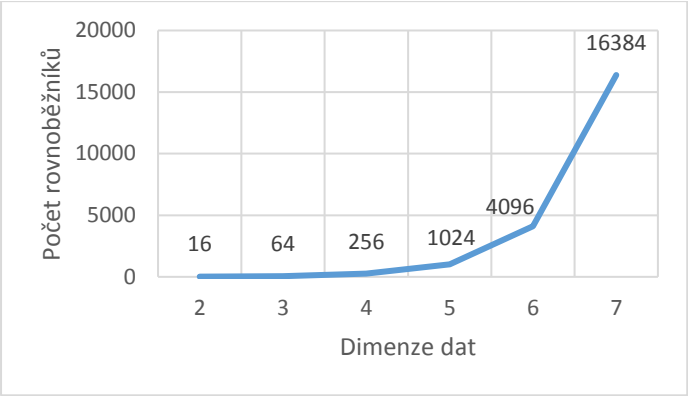
3.1.1 Analýza omezení paralelních množin

Jak již bylo popsáno v kapitole 2.6, technika paralelních množin má odlišné omezení pro stromové a svazkové rozložení.

Stromové rozložení rozkládá vztahy mezi různými atributy na nejjemnější možnou úroveň, což ale pro velký počet dimenzí vede k příliš velkému počtu rovnoběžníků mezi osami a velkému počtu jejich křížení. Tím se vizualizace stává nepřehlednou a pro velký počet dimenzí nepoužitelnou. Nejhorší případ pro počet dělení rovnoběžníků nastane, pokud pro každou kombinaci kategorií existuje položka dat, která v těchto kategoriích leží. V takovém případě lze počet rovnoběžníků mezi posledním párem os vyjádřit jako:

$$n_T = \prod_{1}^{N} C_i$$

kde N je počet os a C_i je počet kategorií *i-té* osy. Pokud bychom uvažovali například čtyři kategorie pro každou osu, počet rovnoběžníků mezi posledním párem os by se zvyšující se dimenzí rostl exponenciálně a pro čtvrtou dimenzi by už mezi posledními osami vedlo 256 rovnoběžníků (viz Obrázek 15).



Obrázek 15: Maximální počet rovnoběžníků mezi posledním párem os pro danou dimenzi

Dalším problémem snižujícím čitelnost je křížení rovnoběžníků. Při stromovém dělení může každý rovnoběžník v oblasti mezi párem sousedních os křížit ostatní, které vycházejí ze stejné kategorie i ty, které vycházejí z jiných kategorií. Maximální počet křížení mezi posledními dvěma osami lze vyjádřit jako maximální počet křížení v kompletním bipartitním grafu [29]:

$$X_T = \left\lfloor \frac{\prod_1^{N-1} C_i}{2} \right\rfloor \left\lfloor \frac{\prod_1^{N-1} C_i - 1}{2} \right\rfloor \left\lfloor \frac{C_N}{2} \right\rfloor \left\lfloor \frac{C_N - 1}{2} \right\rfloor$$

N je počet os, C_i je počet kategorií *i-té* osy a C_N je počet kategorií poslední osy.

Počet křížení s rostoucí dimenzí tedy roste ještě rychleji než počet rovnoběžníků. Pro vzorový příklad se čtyřmi kategoriemi na každé ose, by pro čtvrtou dimenzi mezi posledním párem os mohlo být až 1984 křížení rovnoběžníků.

Při použití stromového rozložení tedy počet rovnoběžníků i počet křížení s rostoucí dimenzí roste velmi rychle. Oproti tomu při použití svazkového rozložení je počet rovnoběžníků mezi párem os v nejhorším případě následující:

$$n_B = c \,.\, C_i \,.\, C_{i+1}$$

kde *c* je počet kategorií na první ose vlevo, C_i je počet kategorií na levé ose páru a C_{i+1} je počet kategorií na pravé ose páru.

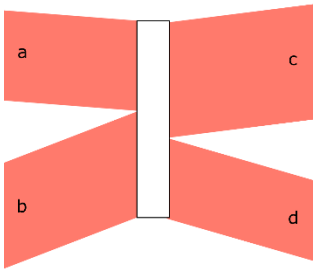
Co se týče křížení, tak ve svazkovém rozložení se nemohou křížit rovnoběžníky vycházející ze stejné kategorie, ale stále se mohou křížit ty, které vycházejí z různých kategorií. Maximální počet křížení mezi každým párem sousedních os je potom následující:

$$X_B = \left\lfloor \frac{C_i}{2} \right\rfloor \left\lfloor \frac{C_i - 1}{2} \right\rfloor \left\lfloor \frac{C_{i+1}}{2} \right\rfloor \left\lfloor \frac{C_{i+1} - 1}{2} \right\rfloor$$

C_i je počet kategorií na levé ose páru a C_{i+1} je počet kategorií na pravé ose páru.

Z uvedených vztahů je zřejmé, že ve svazkovém rozložení je mnohem menší počet rovnoběžníků mezi páry os a dochází i k menšímu počtu křížení. Počet rovnoběžníků je dokonce oproti stromovému rozložení nezávislý na počtu dimenzí. Tím pádem je svazkové rozložení i pro velký počet dimenzí přehledné a čitelné, a je tedy pro data s větším počtem dimenzí vhodnější. Toto rozložení má ale zase jiné omezení, které již bylo zmíněno v sekci 2.6. Sledování závislostí mezi

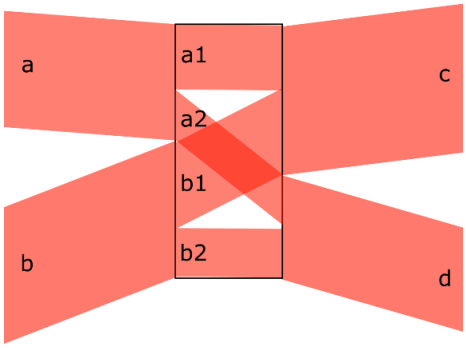
jednotlivými dimenzemi je složitější a v některých případech nemusí být možné. Jedná se o případ, kdy do jedné kategorie vstupuje dva nebo více rovnoběžníků stejné barvy a opět dva nebo více rovnoběžníků téže barvy vystupuje (Obrázek 16). V takovém případě nelze jednoznačně určit v jakém poměru se vstupní rovnoběžníky ***a*** a ***b*** dělí do výstupních ***c*** a ***d***.



*Obrázek 16: Problém svazkového rozložení – do jedné kategorie vstupují rovnoběžníky **a, b** stejné barvy a **c, d** téže barvy vystupují*

3.1.2 Popis metody Set Rivers

Základní myšlenka, jak řešit výše popsané problémy, je v použití svazkového rozložení, ale upraveného tak, aby nevznikaly nejednoznačné případy a vždy šlo určit, jak se rovnoběžník dělí a kam pokračuje při průchodu osou. Toho je docíleno tak, že jsou přidány nové rovnoběžníky do prostoru obdélníku, který ohraničuje kategorii. Tyto rovnoběžníky propojují vstupní a výstupní vazby kategorie způsobem, který lze jednoduše demonstrovat na obrázku níže (Obrázek 17). Vstupní rovnoběžník ***a*** se uvnitř kategorie dělí na nově vzniklé ***a1*** a ***a2***, kde výška ***a1*** odpovídá poměru dat, které z rovnoběžníku ***a*** pokračují do rovnoběžníku ***c*** a výška ***a2*** odpovídá poměru dat, které pokračují do ***d***. Součet výšek ***a1*** a ***a2*** odpovídá výšce ***a***, tedy množství dat, které reprezentuje celý rovnoběžník ***a***. Stejným způsobem se dělí rovnoběžník ***b*** na ***b1*** a ***b2***, podle toho, kolik dat pokračuje do rovnoběžníku ***c*** a ***d***. Takto lze z vizualizace určit, jaké poměry dat kam pokračují a nenastane výše popsaný nejednoznačný případ.



Obrázek 17: Propojení rovnoběžníků uvnitř kategorie

Maximální počet takto vytvořených nových rovnoběžníků uvnitř jedné kategorie popisuje vzorec

$$n_R = q \,.\, C_{i-1} \,.\, C_{i+1}$$

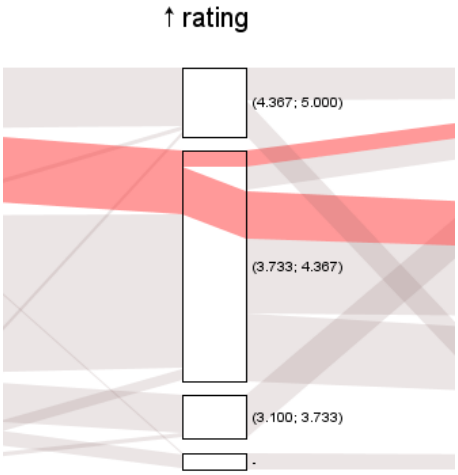
kde C_{i-1} je počet kategorií předchozí osy a C_{i+1} je počet kategorií následující osy. q je parametr určený počtem uživatelem definovaných dotazů, které jsou popsány níže.

Maximální počet křížení rovnoběžníku uvnitř kategorie je dán vzorcem:

$$X_R = q \cdot \left\lfloor \frac{C_{i-1}}{2} \right\rfloor \left\lfloor \frac{C_{i-1}-1}{2} \right\rfloor \left\lfloor \frac{C_{i+1}}{2} \right\rfloor \left\lfloor \frac{C_{i+1}-1}{2} \right\rfloor$$

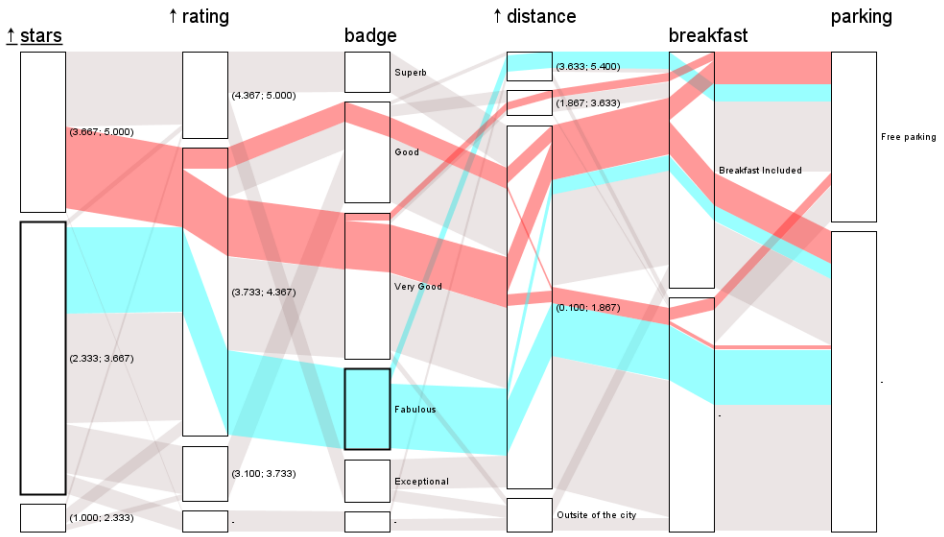
Tímto způsobem vznikne uvnitř kategorie křížení rovnoběžníků dané kategorie a mezi osami se kříží jen rovnoběžníky z různých kategorií. Prostory pro křížení jsou tedy oddělené na rozdíl od stromového rozložení. Ve stromovém rozložení oba tyto typy křížení probíhají v prostoru mezi osami, což vede k menší přehlednosti. Cenou na toto zjednodušení je ale malý prostor uvnitř kategorie, ve kterém musí být křížení realizováno. Z toho důvodu je vhodné dát uživateli možnost kategorii interaktivně rozšiřovat a zužovat, aby si mohl podrobně prohlédnout i vazby uvnitř kategorie.

Dalším vylepšením této metody je umožnění uživateli filtrovat data podobným způsobem, jako u paralelních souřadnic. Pomocí filtrování uživatel oddělí data, která ho zajímají, a vizualizace může být přehlednější. Vzhledem k diskrétnímu charakteru dat, filtrování neprobíhá výběrem intervalů na osách, jako u paralelních souřadnic, ale jsou vybírány jednotlivé kategorie. Množiny dat z těchto kategorií jsou spojeny pomocí množinových operací a definují výslednou množinu dat, která je poté ve vizualizaci zvýrazněna. Kromě barevného zvýraznění výsledné množiny je zlepšena přehlednost vizualizace ještě tím, že pro ostatní data nejsou vykreslovány rovnoběžníky uvnitř kategorií (Obrázek 18).



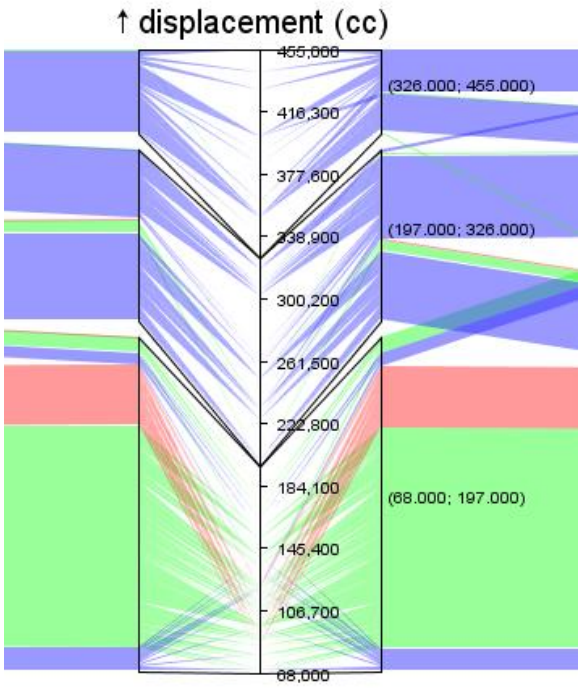
Obrázek 18: Zobrazení rovnoběžníků uvnitř kategorie při použití filtrů

Uživatel může definovat i více množin zároveň. Jednotlivé množiny jsou potom ve vizualizaci zvýrazněny odlišnými barvami (Obrázek 19).



Obrázek 19: Dva filtry zobrazené zároveň

Dalším problémem paralelních množin, na který se metoda Set Rivers zaměřuje, je zobrazení spojitých dat. V běžné vizualizaci pomocí paralelních množin je třeba atributy, které nabývají spojitých hodnot, rozdělit na kategorie a zobrazit stejným způsobem jako kategorická data. Potom ale není možné sledovat konkrétní hodnoty dat. Metoda Set Rivers tento problém řeší možností zobrazit osy spojitých atributů podobným způsobem jako u metody paralelních souřadnic. Toto zobrazení vypadá tak, že uprostřed kategorické osy je vykreslena číselná osa. Rovnoběžníky, které vstupují do kategorií této osy, se uvnitř změní na trojúhelníky, a ukazují na konkrétní hodnotu na číselné ose (Obrázek 20). Každý trojúhelník v tomto způsobu zobrazení odpovídá jedné položce dat.



Obrázek 20: Zobrazení číselné osy uvnitř kategorií

3.2 Požadavky na systém

Cílem této práce je navrhnout a implementovat aplikaci pro vizualizaci heterogenních n-rozměrných dat, která bude splňovat následující požadavky:

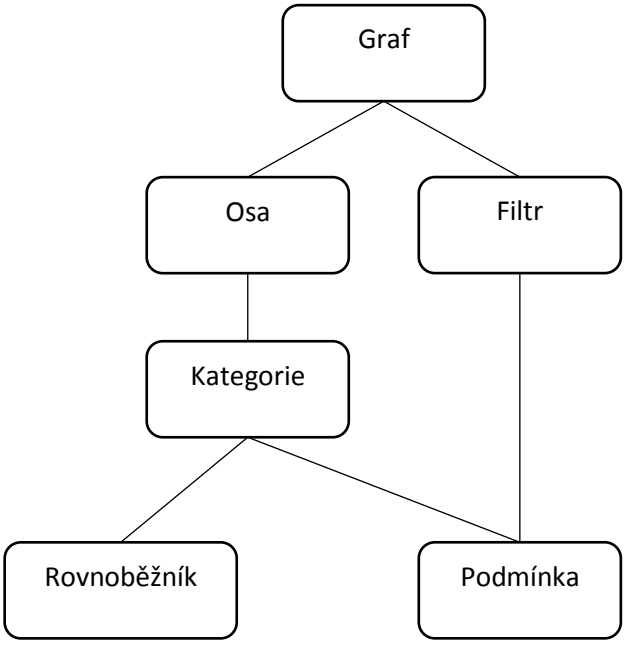
- Navrhovaná aplikace bude umožňovat vizualizaci dat metodou paralelních množin i metodou Set Rivers
- Data budou načítána ze souboru formátu CSV nebo textového souboru
- V rámci metody paralelních množin bude aplikace umožnovat dva režimy rozložení rovnoběžníků: stromové rozložení a svazkové rozložení
- Osy grafu pro spojité atributy budou nejdřív automaticky rozděleny na kategorie, a poté bude mít uživatel možnost kategorie editovat
- Uživatel bude mít možnost určit libovolnou osu jako aktivní a podle kategorií této osy se bude řídit barva rovnoběžníků
- Barvy budou automaticky vygenerovány tak, aby splňovaly pravidla pro kódování informace do barevného kanálu při vizualizaci, a poté je uživatel bude moci změnit podle svých potřeb
- Uživateli bude umožněno interaktivně měnit pořadí os ve vizualizaci
- Uživateli bude umožněno interaktivně měnit pořadí kategorií na osách
- Uživatel bude mít možnost určit, pro které parametry dat budou ve vizualizaci vykresleny osy
- Uživateli bude umožněno interaktivně měnit šířku os a vzdálenost os mezi sebou
- Při zapnutém režimu stromového rozložení paralelních množin bude mít uživatel možnost vytvořit vlastní osu, která bude realizovat techniku skládání dimenzí
- V rámci metody Set Rivers bude aplikace umožní zobrazení rovnoběžníků uvnitř kategorií os
- Pro osy, reprezentující spojité atributy, bude aplikace v rámci metody Set Rivers umožnovat zobrazení číselné osy uvnitř kategorické
- V rámci metody Set Rivers bude mít uživatel možnost filtrovat data
- Aplikace bude umožňovat zobrazení více než jednoho filtru zároveň

3.3 Navrhovaná struktura aplikace

Aplikace je rozdělena do několika částí tak, že každá část zajišťuje určitý logický celek. Hlavní časti jsou tři. První část reprezentuje datový model, kde jsou uchovány položky dat určených pro vizualizaci a informace o atributech dat. Tato část poskytuje informace o datech ostatním částem systému a zajišťuje práci s jednotlivými datovými položkami.

Druhá část zajišťuje vykreslení vizualizace a stará se o uživatelské rozhraní. Kromě vykreslení grafu je zde zajištěno i vykreslení všech komponent grafického uživatelského rozhraní a jejich obsluha v podobě zpracování a předání informace o interakci uživatele do ostatních částí systému.

Poslední část reprezentuje model grafu paralelních množin a zajišťuje operace, které je třeba s grafem v průběhu vizualizace provádět. Model grafu se skládá z prvků, které reprezentují jednotlivé části grafu a zajišťují s nimi spojenou funkcionalitu. Tyto prvky jsou navzájem propojené tak, jak ukazuje následující diagram (Obrázek 21). Propojení odpovídá hierarchii jednotlivých částí grafu.



Obrázek 21: Model grafu paralelních množin

Na vrcholu hierarchie je graf. Tento prvek uchovává obecné informace o celém grafu, jako horizontální vzdálenost os od sebe, šířka os a režim vykreslování (stromové, svazkové rozložení nebo režim Set Rivers). Dále tento prvek uchovává hlavní části grafu, kterými jsou osy a filtry.

Osa uchovává svůj název, rozměry, způsob řazení hodnot na ose a kategorie osy. Zajišťuje řazení svých kategorií a výpočet jejich rozměrů.

Kategorie představuje kategorii hodnot na ose grafu a uchovává svůj název, barvu, rozměry, rovnoběžníky, které touto kategorií procházejí, a podmínku definující jaká data do této kategorie spadají.

Rovnoběžník představuje propojující rovnoběžník v grafu paralelních množin, který spojuje vždy dvě kategorie sousedních os. Tento prvek uchovává rozměry rovnoběžníku, kategorie, které spojuje, a data, která rovnoběžník reprezentuje. Při stromovém rozvržení obsahuje ještě nadřazený rovnoběžník, ze kterého vychází, a podřazené rovnoběžníky, do kterých se dělí.

Filtr uchovává svoji barvu a podmínku, která definuje, jaká data do filtru spadají.

Podmínka je prvek definující určitou množinu dat podle zadaných pravidel. V případě spojitých atributů je pravidly definována horní a spodní hranice hodnot, mezi kterými musí data ležet. Pro kategorické atributy je přímo definovaná hodnota, kterou musí data nabývat. Definice množiny

podmínky může být i spojení jiných takto definovaných množin pomocí množinových operací průniku a sjednocení.

3.4 Návrh procesů

Tato kapitola popisuje návrh a posloupnosti procesů systému, které vedou ke splnění stanovených požadavků. U procesů jsou popsány algoritmy a způsob řešení daných problémů. Pokud je součástí procesu interakce uživatele, jsou popsány a vysvětleny i jednotlivé akce uživatele.

Životní cyklus navrhované aplikace lze popsat jako posloupnost následujících základních procesů:

1. Načtení dat
2. Vytvoření vizualizace
3. Interakce uživatele
4. Aktualizace vizualizace

Data jsou nejprve načtena z textového formátu a prezentována uživateli v podobě tabulky. Přitom také dochází k předzpracování dat tak, aby byla připravena pro vizualizaci paralelních množin. Po tomto kroku následuje fáze, kdy může uživatel data editovat. Může odstraňovat vybrané položky dat, odstraňovat atributy nebo i editovat hodnoty dat. Po ukončení editace zvolí možnost vykreslení grafu paralelních množin a systém pokračuje dalším krokem, tedy tvorbou vizualizace.

Proces tvorby vizualizace se skládá z posloupnosti několika dílčích kroků, které je nutno vykonat. Nejdřív je třeba podle atributů dat definovat osy grafu a ty poté rozdělit na kategorie. Každá kategorie musí mít definovaná data, která do ní spadají, a musí být určeny i rozměry a pořadí kategorií na ose. Po tvorbě os a kategorií následuje tvorba propojujících rovnoběžníků. Tyto rovnoběžníky jsou vypočítány na základě dat a pro každý je určena jeho pozice, rozměry a barva. Když jsou rovnoběžníky vypočítány, je už možné vizualizaci vykreslit.

Po vykreslení vizualizace následuje fáze, kdy systém čeká na interakci od uživatele. V této fázi může uživatel měnit aktivní osu grafu, přesouvat, zobrazovat a skrývat osy, přesouvat a editovat kategorie, měnit šířku os, vytvářet vlastní osu nebo filtrovat data. Po každé z těchto interakcí musí dojít k přepočítání grafu a novému vykreslení aktualizované vizualizace. Aktualizace s novým vykreslením probíhá ihned po každé akci uživatele, při které došlo ke změně grafu, takže uživatel má při interakci okamžitou zpětnou vazbu. Zároveň jsou uchovávány údaje o typu změny, podle kterých je poté při aktualizaci rozhodnuto, jaké parametry grafu musí být přepočítány. Při každé aktualizaci tedy nedochází k počítání celého grafu, ale jen potřebné části.

Dále v této kapitole jsou uvedené procesy blíže popsány.

3.4.1 Načtení dat

Aplikace umožňuje načíst vstupní data pro vizualizaci ze souboru CSV nebo jakéhokoliv textového souboru, který splňuje definované formátování textu:

- První řádek obsahuje hlavičku tabulky – názvy jednotlivých atributů
- Pro ostatní řádky platí, že jeden řádek obsahuje vždy hodnoty pro jednu položku dat
- Pořadí hodnot v řádku odpovídá pořadí atributů v hlavičce
- Názvy atributů, stejně jako jednotlivé hodnoty dat, jsou odděleny oddělovačem. Jako oddělovač může sloužit jeden z následujících znaků: mezera, tabulátor, čárka, středník nebo libovolný definovaný znak. V celém souboru ale musí být oddělovač stejný a nesmí se vyskytovat v názvech atributů nebo v hodnotách.
- Před hodnotou dat může být uveden kód pro řazení hodnot, který se používá v případě nominálních atributů. Kód začíná speciálním znakem „#“, následuje číslo, které definuje pořadí hodnoty, a poté mezera, která tento kód odděluje od hodnoty dat.

Při načítání dat probíhá předzpracování, během něhož se určuje typ jednotlivých atributů. Rozlišují se dva základní typy, které se při tvorbě os a kategorií zpracovávají odlišně. Tyto typy jsou: číselný atribut a nečíselný atribut. O typu atributu je rozhodnuto na základě kontroly všech hodnot dat pro daný atribut. Pokud jsou všechny hodnoty číselné, je atribut číselný, pokud není žádná číselná, je nečíselný. V případě, že obsahuje číselné i nečíselné hodnoty, je rozhodnuto na základě počtu nečíselných hodnot. Pokud je nečíselná hodnota jen jedna a zbylé číselné, je atribut označen jako číselný. V ostatních případech je nečíselný. Toto pravidlo ošetřuje případy dat, kde u číselných atributů nemají všechny položky dat definovanou hodnotu. Nedefinované hodnoty jsou pak vyjádřeny libovolným textem typu „undefined“ nebo „-“ anebo je hodnota prázdná. I takové atributy jsou tedy vyhodnoceny jako číselné s tím, že mají jednu speciální kategorii pro nedefinované hodnoty.

3.4.2 Tvorba kategorií

Pro zobrazení grafu paralelních množin je třeba každou osu rozdělit na kategorie hodnot (viz kapitola 2.6). U os, reprezentujících atributy, které nabývají kategorických hodnot, přirozeně odpovídá každá hodnota atributu jedné kategorii osy. Aby bylo ovšem možné v grafu zobrazit i osy, které reprezentují atributy se spojitými číselnými hodnotami, je potřeba i pro tyto osy vytvořit kategorie. Takové kategorie potom obsahují disjunktní intervaly hodnot, které pokrývají celý interval osy. Počet kategorií a jejich spodní a horní hranice ale nejsou nijak předepsány a jejich nastavení záleží na typu dat a cíli uživatele. Proto navrhovaná aplikace při tvorbě grafu nejdřív automaticky vytvoří kategorie pro všechny osy, a poté se tyto kategorie dají uživatelsky upravovat.

Pro číselné osy probíhá automatická tvorba kategorií tak, že je vygenerováno *N* kategorií C_i, kde $i \in \{1, \dots, N\}$ a kategorie C_i pokrývá následující interval hodnot:

$$\langle A_{MIN} + (i-1) \cdot R_C, A_{MIN} + i \cdot R_C)$$

A_{MIN} je minimální hodnota parametru pro danou osu, která se ve vizualizovaných datech vyskytuje, A_{MAX} je maximální hodnota. R_C je rozsah hodnot kategorie, který je dán vztahem $R_C = \lfloor (A_{MAX} - A_{MIN})/N \rfloor$. Počet kategorií *N* je předem nastavená konstanta.

Vzhledem k tomu, že jsou intervaly hodnot kategorií shora otevřené a také vlivem zaokrouhlování při dělení se může stát, že kategorie nepokrývají celý interval hodnot osy. Tento případ je ošetřen přidáním intervalu $\langle i \cdot R_C, A_{MAX} \rangle$ ke kategorii pro nejvyšší hodnoty.

Ke každé kategorii je při vytvoření přiřazen ještě poměr její výšky vůči výšce osy. Tato hodnota je v kategorii uložena a použita při vykreslování, aby nemuselo při každém překreslení docházet k jejímu výpočtu. Hodnota poměru výšky je dána následujícím vztahem:

$$H_i = D_i / D$$

kde D_i je počet položek dat, které procházejí kategorií a *D* je celkový počet položek dat.

Automatická tvorba kategorií číselných os probíhá podle následujícího navrženého algoritmu:

1. R_C je rozsah hodnot kategorie číselné osy *A*, A_{MAX} je maximální hodnota dat na ose *A*, počet kategorií *N* je předem nastavená konstanta
2. Proměnná *F* je nastavena na minimální hodnotu dat na ose *A*
3. Proměnná *T* je nastavena na hodnotu $F + R_C$
4. Pro všechna *i* od 1 do *N* je vytvořena *i-tá* kategorie na ose *A* následujícím způsobem:
    - *a.* Když *i* = *N*
        - i. rozsah *R* je nastaven na hodnotu $\langle F, A_{MAX} \rangle$
    - b. Jinak
        - i. rozsah *R* je nastaven na hodnotu $\langle F, T \rangle$
    - c. C_D je prázdná množina
    - d. *D* je množina všech položek dat
    - e. Pro všechny položky dat D_i z množiny *D*
        - i. Když hodnota D_i pro osu *A* leží v rozsahu *R*
            1. D_i je přidána do množiny C_D
    - f. Výška *H* je spočítána jako *H* = počet prvků v C_D / počet prvků v *D*
    - *g.* Na ose *A* je vytvořena nová kategorie s rozsahem hodnot *R,* množinou dat C_D a relativní výškou *H*
    - h. $F = T$
    - i. $T = T + R_C$

Pro nečíselné kategorické osy je vytvořena jedna kategorie pro každou rozdílnou hodnotu dat na ose. Výška kategorií je vypočítána stejným způsobem jako v případě číselných os.

3.4.3 Řazení kategorií

Po vytvoření kategorií je třeba určit jejich pořadí na ose. Pro číselné osy je pořadí přirozeně určeno hodnotami dat v kategoriích. Díky faktu, že intervaly hodnot kategorií jsou disjunktní, může řadící algoritmus kategorie jednoduše seřadit podle minimální hodnoty intervalu. Jediný

parametr, který je třeba určit, je, jestli bude pořadí kategorií vzestupné nebo sestupné. Vhodnost vzestupného nebo sestupného způsobu řazení záleží na charakteru atributu dat, který osa reprezentuje, a pro různé osy se může lišit. Proto jsou nejdřív při vygenerování grafu všechny číselné osy seřazeny sestupně a následně může uživatel interaktivně způsob řazení měnit.

Pro nečíselné kategorické osy je ale seřazení kategorií složitější. Atributy, které tyto osy reprezentují, mohou být ordinální nebo nominální. To znamená, že buďto nemusí vůbec existovat vhodný způsob řazení, nebo existuje, ale z textového popisu hodnot by bylo velmi složité ho algoritmicky rozpoznat (například slovní vyjádření hodnocení, dny v týdnu apod.). Proto byly pro tyto osy navrženy tři způsoby řazení kategorií. První způsob je, že uživatel pořadí hodnot definuje již v datech před načtením aplikací. V definici je potom jako první znak v hodnotě dat umístěn speciální symbol „#“ a následuje číslo vyjadřující pořadí hodnoty. Tento způsob je vhodný právě pro ordinální atributy, které mají vhodný způsob řazení, ale z názvu hodnot těžko rozpoznatelný. Další způsob řazení je abecední řazení podle názvu hodnoty. Takový způsob je v hodný pro hodnoty, které mají textový popis začínající číslicí, nebo pro nominální atributy, kde žádný vhodný způsob řazení neexistuje. Poslední způsob je ruční seřazení kategorií uživatelem v již vykresleném grafu. Celý proces řazení nečíselných os tedy probíhá následovně:

1. Při prvním generování grafu je otestováno, jestli hodnoty mají předdefinované řazení pomocí speciálního symbolu.
    a. Pokud mají, tak jsou kategorie seřazeny podle předdefinovaných hodnot.
    b. Pokud nemají definované řazení, jsou seřazeny abecedně.
2. Po vykreslení grafu může uživatel interaktivně pořadí kategorií měnit a toto nové pořadí má potom přednost před automatickým řazením.

3.4.4 Tvorba propojujících rovnoběžníků

Po vytvoření os a kategorií je pro vykreslení grafu paralelních množin potřeba vytvořit rovnoběžníky, které osy propojují. Tyto rovnoběžníky představují množství dat spadajících do propojených kategorií způsobem popsaným v kapitole 2.6. Rovnoběžníky vždy propojují kategorie dvou sousedních os a jejich důležité parametry jsou výška rovnoběžníku a barva. Rozmístění rovnoběžníků a jejich parametry musí být vypočítány na základě vizualizovaných dat a režimu rozložení. Pro vizualizační aplikaci byly navrženy dva režimy rozložení, které se používají v metodě paralelních množin. Je to stromové rozložení a svazkové rozložení. Princip těchto režimů spolu s jejich výhodami, nevýhodami a vhodnosti použití je rovněž popsán v kapitole 2.6. Způsob tvorby rovnoběžníků se v těchto dvou režimech liší, proto bude popsán pro každý režim samostatně. Kromě těchto dvou základních režimů paralelních množin byl v aplikaci navržen i způsob propojení rovnoběžníků uvnitř os podle nově navržené metody Set Rivers. Tento způsob je popsán níže v kapitole 3.4.12 Rovnoběžníky uvnitř os.

Proces tvorby rovnoběžníků má za účel pouze vytvoření topologie grafu. Dojde k vytvoření všech rovnoběžníků a pro každý se určí, které kategorie bude spojovat. Přesné umístění v rámci kategorie a rozměry rovnoběžníků jsou spočítány až v procesu aktualizace, který následuje po tvorbě.

Stromové rozložení

Při stromovém rozložení je jedna osa vybrána jako aktivní a tvoří kořen stromu. Rovnoběžníky potom tvoří stromovou strukturu tak, že směrem od kořenové osy, se na každé další ose rozdělí a pokračují do kategorií následující osy. Při tvorbě rovnoběžníků se nejdřív vytvoří rovnoběžníky mezi kořenovou osou A_1 a sousední osou vpravo A_2. Pro všechny kombinace kategorií z těchto os, kde obě kategorie obsahují alespoň jednu společnou položku dat, je vytvořen rovnoběžník. K němu jsou přiřazeny kategorie, které spojuje, a jejich společná data. Vzniknou tedy rovnoběžníky R_i, $i = \{1, \dots, n\}$. Tvorba rovnoběžníků dále pokračuje v sousedním páru os vpravo. To znamená, že levá osa v novém páru je osa A_2 a pravá je její sousední osa vpravo A_3. Pro každý rovnoběžník R_i z předchozího kroku jsou nalezeny kategorie C_i na ose A_3, pro které platí, že existuje alespoň jedna položka dat, která leží zároveň v rovnoběžníku R_i a v kategorii C_i. Pro každou z těchto kategorií je vytvořen rovnoběžník spojující kategorii, v níž končil rovnoběžník R_i, s kategorií C_i. Poté se stejným způsobem pokračuje v páru o jednu osu vpravo, a takto až k poslední ose.

Popsaným způsobem vznikne stromová struktura rovnoběžníků, která začíná v kořenové ose a větví se mezi dalšími osami směrem doprava. V navrhované aplikaci má ale uživatel možnost zvolit jako aktivní kořenovou osu i jinou než první osu vlevo, čímž dojde k větvení rovnoběžníků i směrem doleva od kořenové osy. Tento případ je vyřešen druhým průchodem výše popsaného algoritmu, ale od kořenové osy směrem doleva.

Svazkové rozložení

Tvorba rovnoběžníků pro svazkové rozložení je jednodušší než u stromového rozložení, protože rovnoběžníky netvoří stromovou strukturu napříč grafem, ale mezi jednotlivými páry os jsou nezávislé. Proces tvorby probíhá tak, že pro všechny páry sousedních os jsou postupně vybírány všechny kombinace kategorií z první a druhé osy v páru. Pokud pro danou kombinaci existují data spadající do obou kategorií, jsou tato data ještě rozdělena do podmnožin podle kategorií aktivní osy, které určují barvu rovnoběžníku. Pro každou z těchto podmnožin dat je vytvořen rovnoběžník mezi kategoriemi dané kombinace s barvou podle kategorie aktivní osy.

3.4.5 Výpočet výšky a pozice rovnoběžníků

Při tvorbě rovnoběžníků jsou každému rovnoběžníku přiřazeny položky dat, které reprezentuje, a kategorie os, které spojuje. Přesná výška rovnoběžníku a pozice na ose je ale vypočítána až v procesu aktualizace. Tento proces musí následovat po každé tvorbě nových rovnoběžníků, aby mohly být vykresleny. Od procesu tvorby je ale oddělen, aby mohl být spuštěn i samostatně. Samostatné spuštění procesu aktualizace je třeba provést vždy, když je nutné přepočítat rozměry rovnoběžníků, například po změně rozměrů kategorií nebo os.

Výška rovnoběžníku v grafu paralelních množin znázorňuje množství dat, které do tohoto rovnoběžníku spadá. Výpočet výšky rovnoběžníku R, který vychází z kategorie C ukazuje následující vztah:

$$H_R = H_C \cdot \frac{N_R}{N_C}$$

kde H_C je výška kategorie C, N_R je počet vzorků dat spadajících do rovnoběžníku R a N_C je počet vzorků dat spadajících do kategorie C.

Pro vykreslení rovnoběžníku je ještě nutné spočítat jeho přesnou vertikální pozici v kategoriích na jeho levém i pravém konci. Vertikální pozici horní hrany rovnoběžníku při vstupu do kategorie C, v podobě vzdálenosti od horního kraje vykreslovacího plátna, lze vyjádřit jako:

$$y = y_C + \sum_{1}^{N} H_i$$

kde y_C je vertikální pozice horního kraje kategorie C a H_1 až H_N jsou výšky rovnoběžníků, které jsou v kategorii C umístěny nad vykreslovaným rovnoběžníkem.

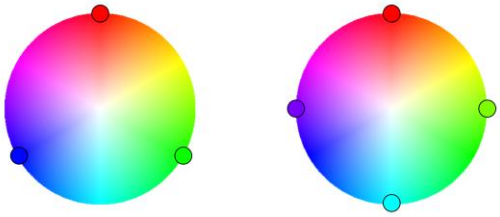
K výpočtu pozice je tedy nutné znát pořadí rovnoběžníků v kategorii, které se liší v závislosti na použitém rozvržení. Pro svazkové rozvržení je pořadí rovnoběžníků v jedné kategorii určeno pořadím kategorií na druhých koncích těchto rovnoběžníků. To znamená, že rovnoběžník, který z jedné kategorie pokračuje na další ose výš než jiný z té samé kategorie, bude i na první ose umístěn výš. Pokud pokračují i na druhé ose do stejných kategorií a liší se pouze barvou, je pořadí určeno pořadím kategorií aktivní osy, které barvy definují. U stromového rozložení má na pořadí rovnoběžníků hlavní vliv pořadí předchozích větví, z nichž rovnoběžníky vycházejí. Pokud vycházejí se stejné větve, je uplatněno stejné pravidlo jako u svazkového rozložení.

3.4.6 Generování barev

V grafu paralelních množin jsou rovnoběžníky rozlišeny podle barev. Každá kategorie vybrané aktivní osy má svoji definovanou barvu a touto barvu jsou obarveny všechny rovnoběžníky, které reprezentují data spadající do dané kategorie. Napříč celým grafem lze tedy jednoduše sledovat, do jaké kategorie aktivní osy rovnoběžník patří.

Aby obarvení splnilo svůj účel, musí být respektována základní pravidla pro kódování informace do barev ve vizualizaci. V případě rozlišení kategorií aktivní osy jde o kódování diskrétních hodnot do barevného vizuálního kanálu. Při takovém kódování je třeba dodržet, aby jednotlivé hodnoty vizuálního kanálu byly od sebe snadno rozlišitelné [18]. V barevném vizuálním kanálu je tedy třeba zvolit takové barvy, aby rozdíl v jejich vjemu byl co největší. Proto byl pro výběr barev zvolen barevný model HSL, který odpovídá lidskému vjemu barev více než model RGB [11].

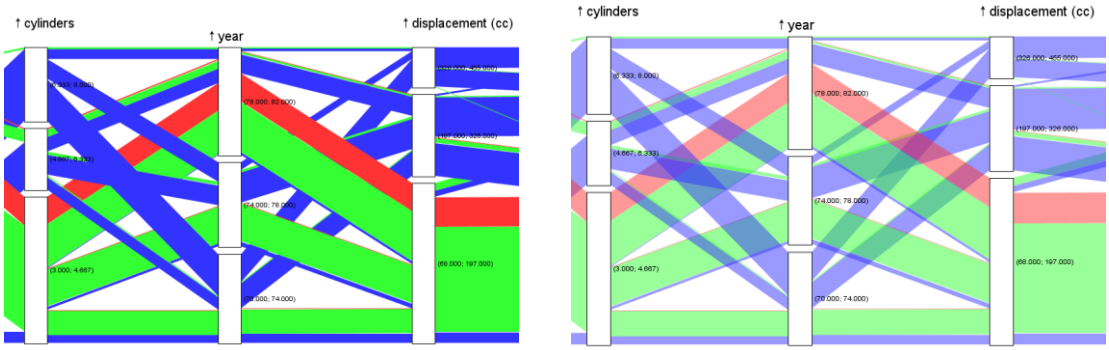
Při procesu generování barev pro rovnoběžníky je pro vygenerování N barev vybráno N bodů z barevného prostoru HSL. Vybrané body mají konstantní sytost a jas, a jsou rovnoměrně rozmístěny v rozsahu hodnot odstínu (Obrázek 22).



Obrázek 22: Výběr barev z HSL modelu – tři a čtyři vzorky barev rovnoměrně rozmístěné v rozsahu hodnot odstínu

Tyto vybrané body reprezentují barvy, které mají vždy největší možný rozdíl v odstínu pro daný počet barev, a tím pádem jsou snadno rozlišitelné a vhodné pro použití ve vizualizaci.

Při obarvování rovnoběžníků je třeba se kromě rozlišitelnosti barevných odstínů zaměřit ještě na přehledné zobrazení křížících se rovnoběžníků. V místech křížení se rovnoběžníky překrývají a nemusí být vždy zřetelné, kam který pokračuje. Proto je u barvy rovnoběžníků snížena hodnota alfa kanálu. Všechny rovnoběžníky jsou vykreslovány s průhledností 50 % (Obrázek 23).



(a) *(b)*

Obrázek 23: Vliv průhlednosti na vykreslování rovnoběžníků - (a) průhlednost 0 %, (b) průhlednost 50 %

Automatické generování barev probíhá vždy při vytváření nového grafu nebo při změně aktivní osy tak, aby byly zvoleny co nejhodnější barvy. Uživatel ale má možnost barvy později interaktivně změnit podle vlastní potřeby.

3.4.7 Vykreslení grafu

Po vytvoření os, kategorií a rovnoběžníků a výpočtu jejich pozic a rozměrů následuje samotné vykreslení grafu. Při procesu vykreslení se již nepřepočítávají hodnoty z dat, ale použijí se již vypočítané, které jsou přiřazeny k daným objektům. Proces vykreslování totiž může proběhnout několikrát beze změny grafu a přepočítávat všechny hodnoty by bylo zbytečné a neefektivní. Samotné vykreslení tedy proběhne jednoduchým průchodem všech rovnoběžníků, při kterém je každý rovnoběžník vykreslen na základě pozic a barvy, která je k němu uložena, a poté průchodem všech os a jejich kategorií, které jsou i s popisky vykresleny přes rovnoběžníky.

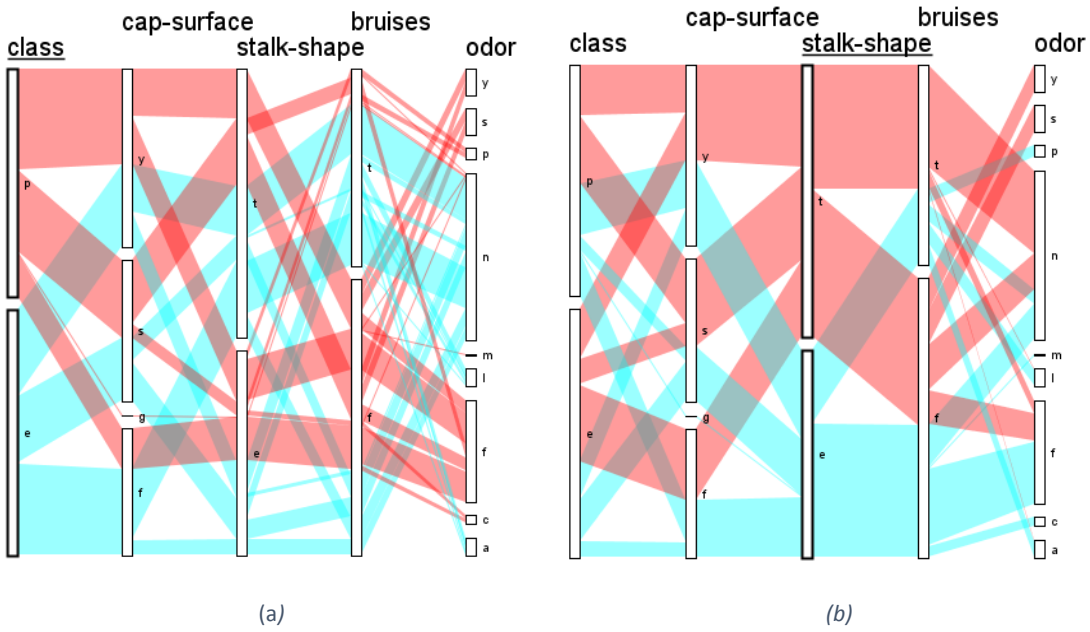
Pokud dojde vlivem interakce uživatele k úpravě grafu, je před dalším vykreslením spuštěn proces aktualizace, v němž jsou potřebné hodnoty přepočítány.

3.4.8 Návrh základních způsobů interakce

S ohledem na požadavky systému byly navrženy způsoby interakce uživatele, které lze rozdělit na základní a pokročilé. Tato kapitola se věnuje základním způsobům interakce, což jsou změna aktivní osy, přesouvání, zobrazování a skrývání os, přesouvání kategorií a změna šířky os a prostoru mezi nimi. Pokročilé způsoby interakce, jako editace kategorií, vytváření vlastní osy pro stromové rozložení a filtrování, jsou popsány samostatně.

Změna aktivní osy

Podle aktivní osy se odbarvují rovnoběžníky v celém grafu a pro stromové rozložení navíc definuje kořen stromu rovnoběžníků. Je tedy vhodné, aby si uživatel mohl zvolit, která osa bude aktivní, v průběhu práce ji mohl lehce měnit a ihned dostávat zpětnou vazbu o změně grafu. Způsob této interakce je realizován kliknutím myši na název osy, kterou chce uživatel zvolit jako aktivní. Ihned po výběru aktivní osy uživatelem dojde k přepočítání rovnoběžníků a vykreslení nové podoby grafu. Informace, která osa je aktuálně vybrána jako aktivní, je uživateli zobrazena v podobě silnějšího obrysu aktivní osy a podtržení jejího názvu (Obrázek 24).



Obrázek 24: Výběr aktivní osy – aktivní osa může být první osa vlevo (a) nebo i kterákoli osa uprostřed (b)

Pokud uživatel vybere při stromovém rozložení jako aktivní osu takovou, která není na kraji grafu (Obrázek 24b), rovnoběžníky se větví od této osy na obě strany.

Přesouvání os a kategorií

Dalším základním způsobem interakce u grafu paralelních množin je změna pořadí os, díky které může uživatel lehce srovnat závislosti mezi jakýmikoliv dvěma osami. Tento způsob interakce je realizován technikou drag-and-drop, kde uživatel myší přetáhne požadovanou osu na potřebné místo. Během tažení myší je přes graf vykreslována poloprůhledná šedá čára, indikující, na které místo se osa přesune. Po puštění tlačítka myši dojde k přesunu osy, aktualizaci a překreslení grafu.

Podobným způsobem lze přesouvat i kategorie os, které nejsou číselné. Tyto kategorie mohou reprezentovat nominální hodnoty, u nichž neexistuje žádné vhodné řazení. Proto je vhodné nechat seřazení kategorií na uživateli. V tomto případě probíhá interakce stejně jako u přesouvání kategorií.

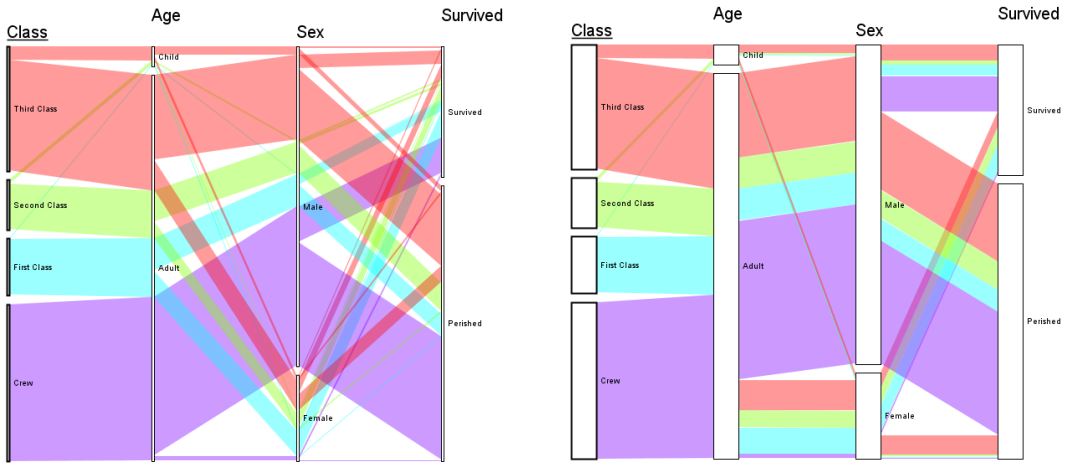
U os, které reprezentují číselné atributy, je pořadí kategorií pevně dáno, ale uživatel může interaktivně ovlivnit, zda budou seřazeny vzestupně nebo sestupně. To lze provést klinutím na ikonu šipky před názvem osy. Pomocí této ikony lze také identifikovat, které osy byly vyhodnoceny jako číselné. U nečíselných os se tako ikona nezobrazuje.

Zobrazování a skrývání os

Pro přehlednost vizualizace si uživatel může skrýt osy, které nepotřebuje sledovat. Tuto interakci lze provést dvěma způsoby. První způsob je zobrazení kontextového menu osy kliknutím pravým tlačítkem myši na název osy, kterou chce uživatel skrýt, a výběr možnosti *Hide axis*. Druhý způsob je rozbalení menu se všemi parametry dat kliknutím na tlačítko *Parameters* v horní liště okna s grafem. V tomto menu lze odškrtnutím parametru skrýt osu, která ho reprezentuje a zaškrtnutím ji znovu zobrazit ve vizualizaci.

Změna šířky os a prostoru mezi nimi

Uživateli je také umožněno interaktivně měnit šířku obdélníků, které reprezentují kategorie os. Tuto možnost je vhodné uživateli poskytnout, protože v některých případech je vizualizace přehlednější s užšími osami a v jiných naopak s širšími. Například pro stromové rozložení je vhodné úzké zobrazení os, při kterém je lépe vidět, kam který rovnoběžník pokračuje (Obrázek 25a). Naopak ve svazkovém rozložení rovnoběžníky na osách přímo nepokračují, takže širší prostor os nevadí (Obrázek 25b). Rozšíření os je ale vhodné zejména pro zobrazení rovnoběžníků uvnitř os při metodě Set Rivers.



(a) *(b)*

Obrázek 25: Šířka os

Kromě šířky obdélníků os může uživatel interaktivně měnit i šířku prostoru mezi osami. To je vhodné především pro přizpůsobení vizualizace s různým počtem os velikosti monitoru.

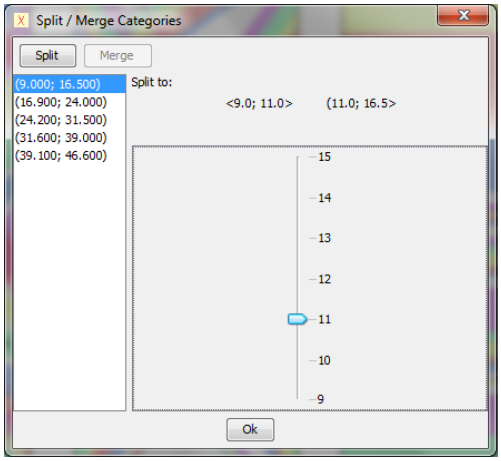
Změna šířky prostoru mezi osami je realizována rolováním kolečka myši. Při rolování a současném držení klávesy *Shift* dochází ke změně šířky os.

3.4.9 Editace kategorií

Mezi pokročilé techniky interakce, které byly v aplikaci navrženy, patří editace kategorií os. Po vygenerování vizualizace, se u spojitých číselných os automaticky vytvoří kategorie rozdělením osy na *N* rovnoměrných disjunktních intervalů. Tyto kategorie, ale nemusí odpovídat potřebám uživatele, proto je mu umožněno automaticky vytvořené kategorie na číselných osách editovat.

Editace kategorií zahrnuje dvě základní akce. První je spojení dvou a více existujících kategorií do jedné nové kategorie a druhá rozdělení existující kategorie na dvě nové v určeném místě. Pomocí těchto dvou akcí je uživatel schopen vytvořit libovolný počet kategorií s libovolnými rozsahy v rámci celého rozsahu hodnot osy.

Interakce probíhá tak, že uživatel kliknutím pravého tlačítka myši na název osy otevře kontextové menu, ze kterého vybere možnost *Categories* a *Split / merge categories*. Touto volbou se otevře dialogové okno, v němž je možné provádět editaci kategorií zvolené osy (Obrázek 26). Toto okno obsahuje v levé části seznam kategorií osy a v pravé rozsah hodnot vybrané kategorie s posuvníkem. Vybráním kategorie ze seznamu, nastavením posuvníku na požadovanou hodnotu a kliknutím na tlačítko *Split* lze kategorii v libovolném místě rozdělit. Spojení kategorií se provádí výběrem dvou a více navazujících kategorií ze seznamu a kliknutím na tlačítko *Merge*.

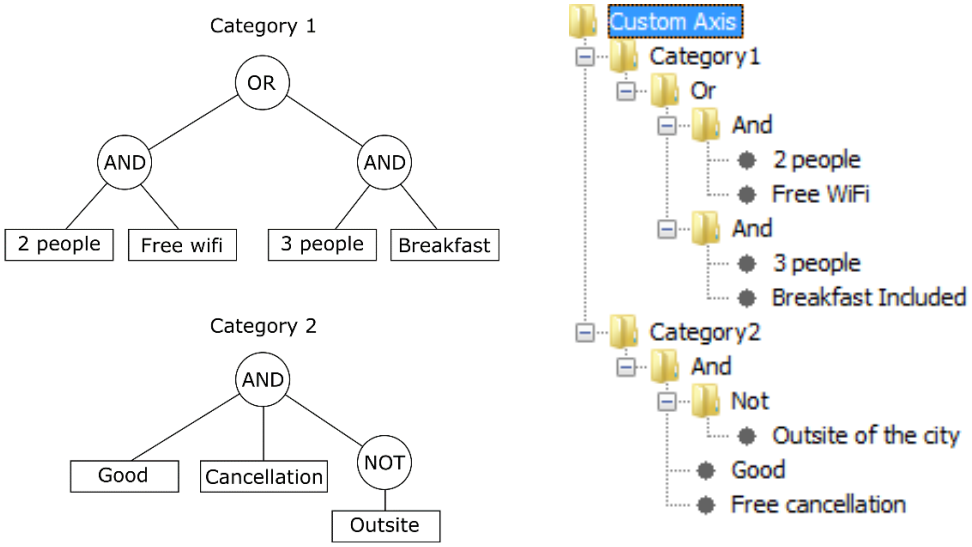


Obrázek 26: Dialogové okno pro editaci kategorií

3.4.10 Vlastní osa ve stromovém rozložení

V přehledu interaktivních technik pro paralelní množiny (kapitola 2.6) byla popsána technika skládání dimenzí. Tato technika umožňuje uživateli vytvořit novou dimenzi založenou na kategoriích ostatních dimenzí v grafu. Pro novou dimenzi potom vznikne nová osa, obsahující klasifikaci dat podle několika jiných os, takže se potom počet zobrazených os v grafu může snížit. Tato technika je v navržené aplikaci realizována pomocí tvorby vlastní osy.

Kategorie vlastní osy si uživatel definuje spojováním existujících kategorií z ostatních os pomocí logických spojek AND, OR a NOT. Tyto spojky představují množinové operace nad množinami dat z kategorií. Spojením kategorií C_1 AND C_2 tedy vznikne kategorie C_3, která obsahuje průnik množin dat z kategorií C_1 a C_2. Podobně C_1 OR C_2 představuje sjednocení a NOT C_1 doplněk C_1 ve všech datech. Operace lze libovolně skládat a vnořovat do sebe, což je ukázáno na následujícím příkladu (Obrázek 27).



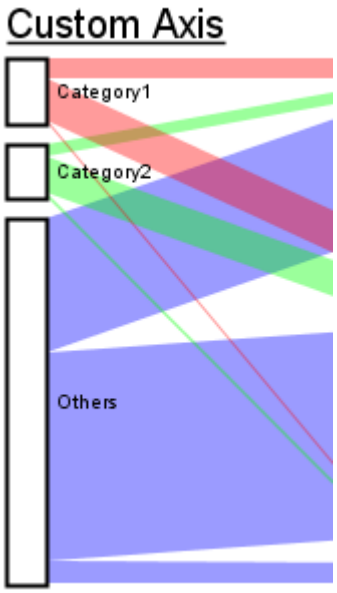
(a) *(b)*

Obrázek 27: Tvorba vlastní osy – spojování kategorií

Zde má uživatelem vytvořená osa dvě kategorie. Kategorie *Category1* obsahuje ta data, která spadají do kategorií *2 people* a zároveň *free* W*iFi* anebo do kategorií *3 people* a zároveň *Breakfast Included*. Kategorie *Category2* obsahuje data, která spadají do kategorií *Free cancelation*, *Good* a zároveň nespadají do *Outside of the city*.

Tvorbu vlastní osy uživatel zahájí tak, že klikne na tlačítko *Create custom axis* v pravém postranním panelu hlavního okna s grafem. Tím se otevře dialogové okno pro tvorbu vlastní osy. Zde výběrem možnosti *Category*, *Add new category* vytvoří novou kategorii, která se zobrazí v náhledu vytvářené osy. Náhled má podobu stromu, kde je kořenem nová osa, uzly další úrovně jsou vytvořené kategorie a pod nimi je definice dat kategorií (Obrázek 27b). Definice dat má podobu existujících kategorií (listy stromu), které mohou být pospojovány logickými spojkami. Po označení uzlu s názvem vytvořené kategorie, může uživatel do kategorie přidat logickou spojku nebo přímo existující kategorii. Existující kategorie je přidána kliknutím na *Condition*, *Set condition*, *Axis category condition* a poté výběrem požadované osy a kategorie. Logická spojka je přidána stejným způsobem, pouze místo *Axis category condition* uživatel zvolí *Logical condition* a poté vybere *And*, *Or* nebo *Not*. Do logických spojek se poté opět stejným způsobem přidá existující kategorie nebo další logická spojka.

Tímto způsobem vznikne nová osa, obsahující uživatelem definované kategorie a kategorii *Others* (Obrázek 28). *Others* je kategorie, která je vygenerována automaticky při vytvoření uživatelské osy a spadají do ní všechna ostatní data, která nespadají do žádné z uživatelských kategorií.



Obrázek 28: Osa vytvořená uživatelem

3.4.11 Filtry

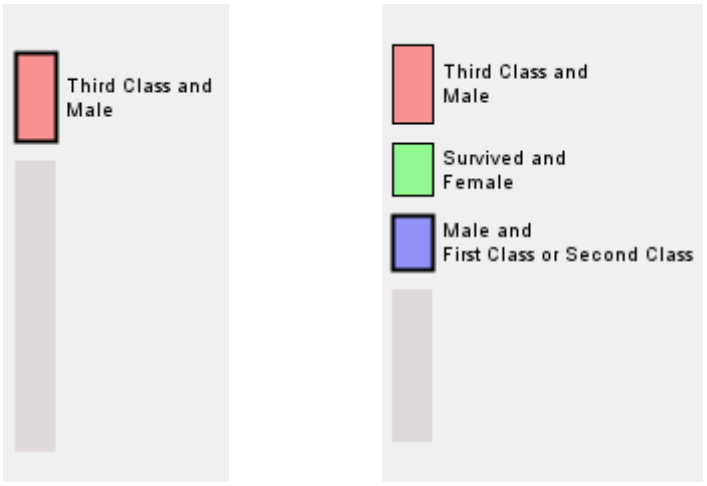
Navrhovaná aplikace umožňuje kromě standardní vizualizace paralelních množin také novou variantu Set Rivers. Jak již bylo nastíněno v kapitole 3.1, jedno z vylepšení, které tako varianta přináší, je možnost interaktivně filtrovat data podobným způsobem jako u paralelních souřadnic. Tato interakce probíhá tak, že uživatel klikáním myší označuje kategorie, které obsahují data v oblasti jeho zájmu. Z vybraných kategorií je vytvořena cílová množina dat, a ta je poté barevně zvýrazněna v grafu. Tato množina je vytvořena následujícím způsobem:

1. A je množina všech os zobrazených ve vizualizaci.
2. Pro každou osu A_i z množiny A je vytvořena množina dat S_i následujícím způsobem:
    a. $S_i = \emptyset$, C je množina všech vybraných kategorií osy A_i
    b. Pro každou kategorii C_i z C je množina dat z C_i přidána do množiny S_i použitím množinové operace sjednocení
3. Výsledná množina S je vytvořena jako průnik všech množin S_i

Aby bylo možné data z vybrané množiny barevně zvýraznit, musí dojít k novému vygenerování všech rovnoběžníků takovým způsobem, aby ty, co reprezentují data z cílové množiny, byly odděleny od ostatních. Zvýraznění je realizováno tak, že rovnoběžníky, které reprezentují vybraná data, jsou obarveny vygenerovanou barvou filtru a ostatní rovnoběžníky jsou vykresleny šedě. Generování barvy filtru probíhá stejným způsobem jako u kategorií (viz kapitola 3.4.6).

Kromě barevného zvýraznění rovnoběžníků v grafu je ještě v pravém postranním panelu vykreslena osa filtrů. Tato osa má stejné vlastnosti jako běžné osy paralelních množin. Je tedy rozdělena na kategorie, jejichž výška odpovídá množství dat, které do nich spadá. Kategorie této osy reprezentují data v jednotlivých filtrech, takže z poměrů velikostí těchto kategorií lze jednoduše sledovat množství vyfiltrovaných dat. Pokud je vytvořen jen jeden filtr, obsahuje tato

osa dvě kategorie, kde jedna odpovídá datům z vytvořeného filtru a druhá všem ostatním, které do filtru nespadají (Obrázek 29a). Pokud je filtrů vytvořeno více, osa vždy obsahuje jednu kategorii pro každý filtr, a v případě, že filtry nepokrývají všechna data, obsahuje ještě jednu kategorii pro ostatní data (Obrázek 29b).

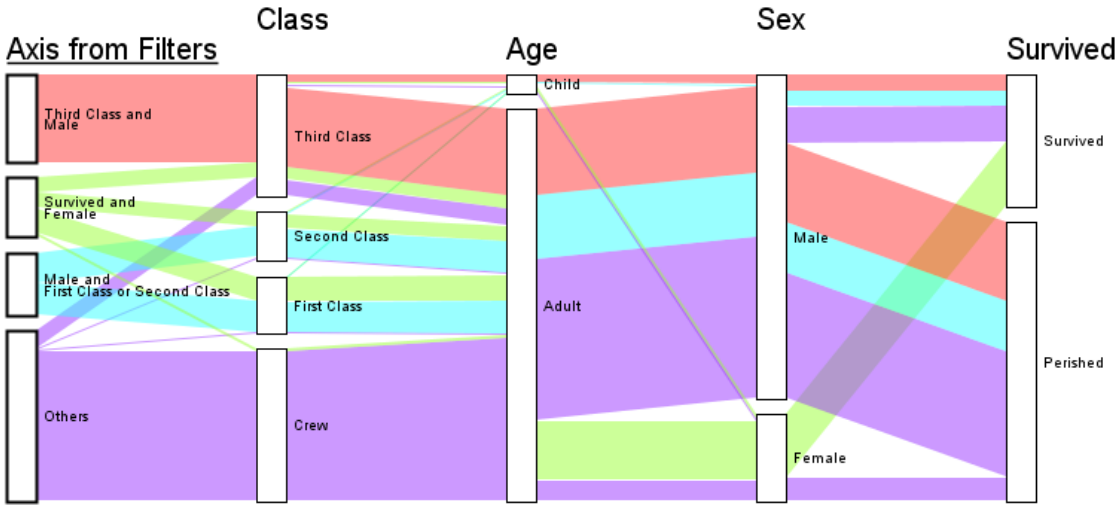


(a) *(b)*

Obrázek 29: Osa filtrů

Pro přidávání nových filtrů a odstraňování existujících slouží ovládací panel filtrů, který je umístěn nad osou filtrů a obsahuje tlačítka *Add new filter* a *Remove selected filter*. Vytvoření nového filtru uživatel provede tak, že klikne na tlačítko *Add new filter* a poté začne vybírat požadované kategorie z grafu. Při vybírání kategorií se vizualizace okamžitě překresluje a uživatel tedy vidí, jak výběr dané kategorie ovlivní výslednou množinu dat. Pro odstranění filtru uživatel vybere daný filtr kliknutím na jeho kategorii na ose filtrů a poté klikne na tlačítko *Remove selected filter*.

Dále má uživatel možnost z osy filtrů vytvořit běžnou osu, která je součástí grafu (Obrázek 30). Tuto akci provede kliknutím na tlačítko *Create axis from filters*. Tím vznikne v grafu nová osa, která je propojena s ostatními osami, a má stejné vlastnosti jako osy reprezentující atributy dat. Její kategorie ale odpovídají datům, které uživatel definoval pomocí filtrů. Vytvořením takové osy tedy uživatel může realizovat techniku skládání dimenzí, jako pomocí vlastní osy v běžné metodě paralelních množin.



Obrázek 30: Osa filtrů jako součást grafu

3.4.12 Rovnoběžníky uvnitř os

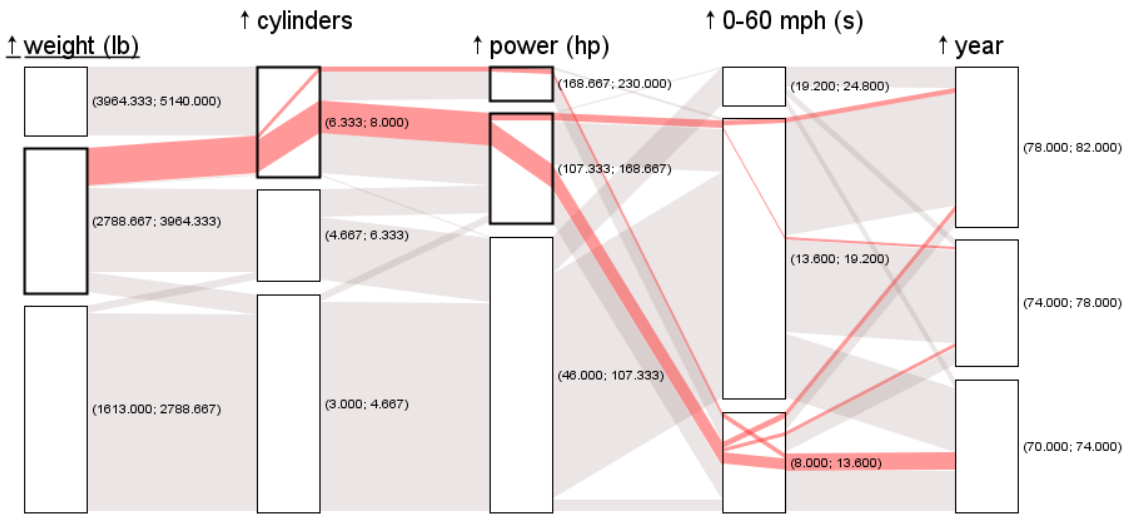
Vizualizační metoda Set Rivers dále umožňuje zobrazit rovnoběžníky uvnitř os, jak bylo popsáno v kapitole 3.1. Princip tvorby těchto vnitřních rovnoběžníků je stejný jako u rovnoběžníků, které propojují osy. Nejdřív je vytvořena topologie, kde je určeno, kolik bude rovnoběžníků v jednotlivých kategoriích a co propojují, a poté je vypočítána pozice a výška pro každý rovnoběžník.

Proces tvorby topologie probíhá podle následujícího algoritmu:

1. C je množina všech kategorií os v grafu
2. Pro každou kategorii C_i z množiny C jsou vytvořeny její vnitřní rovnoběžníky následujícím způsobem:
    a. P_L je množina všech rovnoběžníků, které přicházejí do kategorie C_i zleva, P_R je množina všech rovnoběžníků, které přicházejí do kategorie C_i zprava
    b. Pro každou kombinaci rovnoběžníků P_i, P_j, kde P_i je prvek z množiny P_L a P_J je prvek z množiny P_R, pro kterou platí, že existuje neprázdná množina dat, patřících do rovnoběžníku P_i a zároveň P_j, je vytvořen nový rovnoběžník P_{ij}
    c. Rovnoběžníku P_{ij} jsou přiřazeny rovnoběžníky P_i a P_j, které spojuje, a jako množina dat, kterou rovnoběžník reprezentuje, mu je přiřazen průnik množin dat P_i a P_j

Poté dojde k výpočtu pozice a výšky rovnoběžníků na základě množiny dat, která k nim je přiřazena, a pozice rovnoběžníků, které propojují. Výpočet probíhá stejným způsobem, jak je popsáno v kapitole 3.4.5.

Ve vizualizaci jsou potom uvnitř kategorií vykresleny jen ty rovnoběžníky, které reprezentují data z nějakého definovaného filtru. Rovnoběžníky pro data, která nespadají do žádného z filtrů se uvnitř kategorií nevykreslují, což vede k lepší přehlednosti grafu (Obrázek 31).



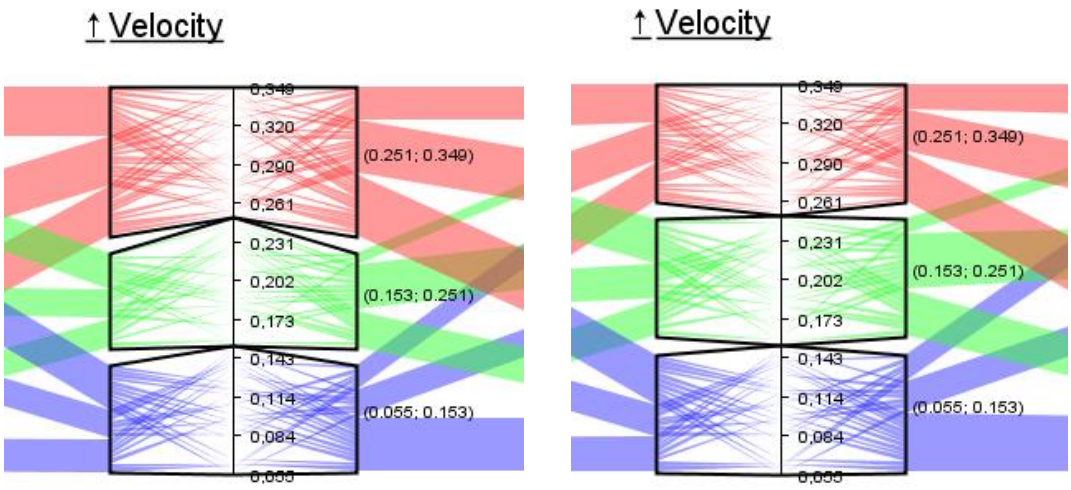
Obrázek 31: Vykreslení rovnoběžníků uvnitř kategorií os

3.4.13 Rozšíření osy o souřadnice

Součástí návrhu aplikace, je také možnost zobrazit číselnou osu uvnitř kategorické, což je další funkcionalita metody Set Rivers. Základní princip spočívá ve vykreslení číselné osy do středu obdélníků tvořících kategorie osy. Rovnoběžníky, které přicházejí na tuto osu, se uvnitř osy změní na trojúhelníky. Každý trojúhelník představuje jednu položku dat a na číselné ose ukazuje na hodnotu, které daná položka dat nabývá.

K vykreslení takové osy je ale možné přistupovat různými způsoby, a proto byly navrženy tři varianty, které lze v aplikaci interaktivně přepínat.

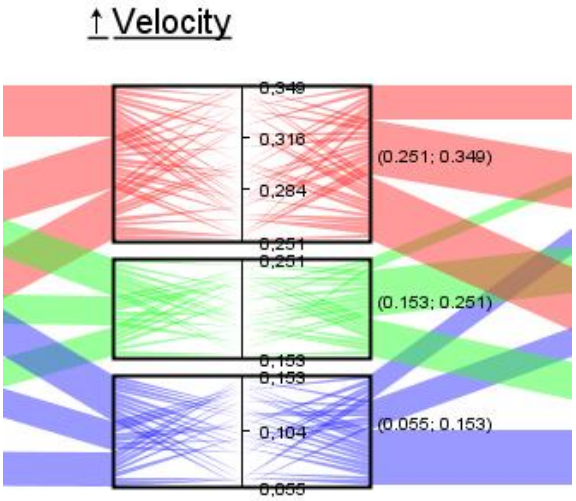
První způsob je, že kategorie osy se řídí běžnými pravidly pro paralelní množiny. Výška a pozice kategorií tedy není závislá na hodnotách dat, ale na množství dat, které do kategorie spadá. Vnitřní číselná osa je spojitá a má lineární rozsah hodnot přes celou výšku grafu bez ohledu na kategorie. Tím pádem pozice horní a dolní hrany obdélníku kategorie na číselné ose neodpovídá maximální a minimální hodnotě dat v kategorii. Proto je třeba hranice kategorie zalomit tak, aby uprostřed ležely na správné hodnotě číselné osy a na krajích odpovídaly správným pozicím kategorie (Obrázek 32a).



(a) *(b)*

Obrázek 32: Lineární číselná osa

V dalším navrženém způsobu se výška kategorií osy přizpůsobí číselné ose. Kategorie na této ose tedy už neodpovídají množství dat, ale hodnotám, které data nabývají. Tím pádem není třeba hrany kategorií zalamovat tak výrazně, jako v předchozím případě, ale zase je třeba rovnoběžníky, které vedly do těchto kategorií, změnit na čtyřúhelníky, které nemají stejnou výšku na levém a pravém konci (rozšiřují se nebo zužují) (Obrázek 32b).



Obrázek 33: Číselná osa rozdělená na kategorie

Poslední variantou je ponechat původní rozměry i obdélníkový tvar kategorií a přizpůsobit číselnou osu. V tomto případě je číselná osa rozdělena podle kategorií a každá její část má jiné měřítko, které odpovídá rozměru a rozsahu hodnot kategorie (Obrázek 33).

4 Implementace

Tato kapitola se věnuje popisu použitých technologií při implementaci aplikace, stručnému popisu systému XDat, který byl při implementaci použit, a přehledu implementovaných tříd.

4.1 Použité technologie

Navržená funkcionalita aplikace pro vizualizaci n-rozměrných dat byla implementována jako rozšíření systému XDat verze 2.2. Pro implementaci byl použit jazyk Java verze 1.8.

4.1.1 XDat

XDat neboli X-dimensional Data Analysis Tool [28] je volně dostupný nástroj pro analýzu vícerozměrných dat šířený pod licencí GPL[1]. Tento nástroj je zaměřený zejména na vizualizaci n-rozměrných dat pomocí metody paralelních souřadnic. Umožňuje načíst data v textovém formátu, zobrazit v tabulce a vizualizovat pomocí metody paralelních souřadnic nebo 2D scatter plot.

Systém je rozdělen na několik částí. Mezi hlavní části patří balíček *chart*, ve kterém jsou třídy reprezentující graf scatter plot, graf paralelních souřadnic a jeho osy a filtry. Dále balíček *data*, který obsahuje reprezentace dat, a balíček *gui*, který obsahuje třídy pro grafické uživatelské rozhraní, včetně tříd *ParallelCoordinatesChartPanel* a *ScatterChart2DPanel*, které zajišťují vykreslení grafu.

Po načtení dat do systému se vytvoří instance třídy *DataSheet*, která obsahuje seznam objektů *Design*, které reprezentují jednotlivé položky dat. Objekty *Design* jsou později použity pro vykreslení lomených čar v grafu paralelních souřadnic, kde jeden objekt *Design* odpovídá jedné lomené čáře v grafu. Stejná reprezentace položek dat byla použita i pro rozšíření systému o vizualizaci dat pomocí paralelních množin. Ke třídě *Design* byla přidána třída *DesignSet*, která reprezentuje množiny dat jednotlivých propojujících rovnoběžníků mezi kategoriemi.

[1] GNU General Public License

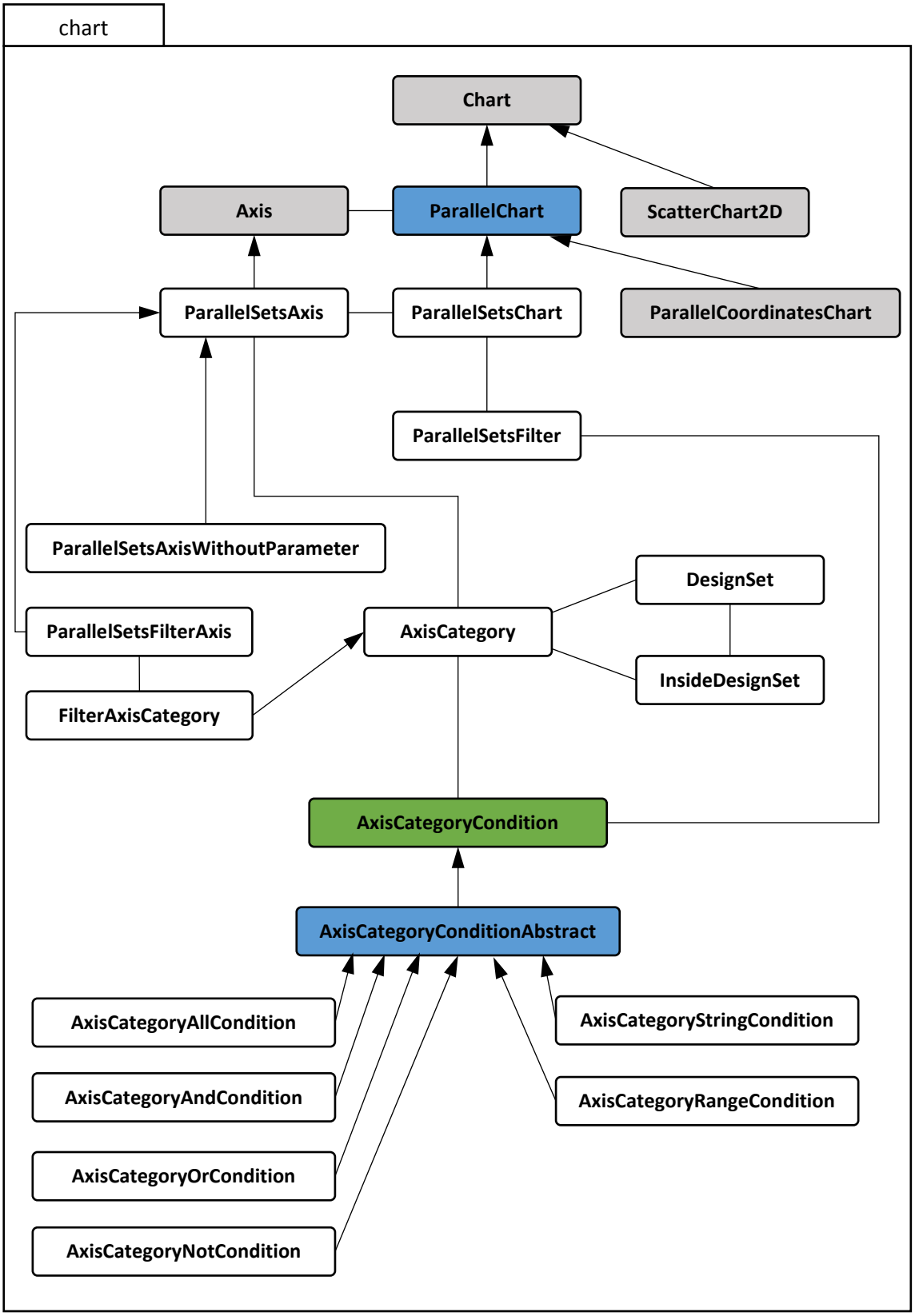
4.2 Struktura tříd

Pro implementaci navržené funkcionality do systému XDat bylo přidáno 32 nových tříd a 41 tříd editováno. Nové třídy jsou rozděleny do balíčků, kde základní balíčky *data*, *gui* a *chart* reflektují základní logické rozčlenění aplikace uvedené v návrhu. Nejvíce tříd bylo přidáno do balíčku *chart*, kde byl doimplementován model grafu paralelních množin.

Struktura tříd balíčku chart je znázorněna na následujícím obrázku (Obrázek 34). Hlavní třídou implementace grafu paralelních množin je třída *ParallelSetsChart*, která rozšiřuje abstraktní třídu Chart, tak, jako původní třídy *ScatterChart2D* a *ParallelCoordinatesChart*. Osy grafu paralelních množin reprezentuje třída *ParallelSetsAxis*, která rozšiřuje třídu Axis z grafu paralelních souřadnic. Třída *ParallelSetsAxis* má dále potomky *ParallelSetsFilterAxis* a *ParallelSetsAxisWithoutParameter*, které reprezentují speciální typy os. *ParallelSetsFilterAxis* reprezentuje osu filtrů vykreslenou v postranním panelu a *ParallelSetsAxisWithoutParameter* reprezentuje uživatelem vytvořenou osu, která na rozdíl od standardních os není přiřazena k žádnému parametru dat. Dále jsou zde třídy reprezentující kategorie os *AxisCategory* a filtry *ParallelSetsFilter*. Třída *AxisCategory* obsahuje odkazy na třídy *DesignSet* a *InsideDesignSet*, které představují propojující rovnoběžníky mezi kategoriemi a uvnitř kategorií. Třídy *AxisCategory* i *ParallelSetsFilter* dále obsahují odkaz na *AxisCategoryCondition*, což je interface, který představuje podmínku definující určitou množinu dat. Tato podmínka byla popsána v návrhu struktury aplikace (kapitola 3.3). Interface *AxisCategoryCondition* má několik implementací, pomocí kterých lze definovat požadovanou množinu dat. Základní jsou *AxisCategoryStringCondition*, která definuje data podle jedné konkrétní nečíselné hodnoty, a *AxisCategoryRangeCondition*, která definuje data podle rozsahu číselných hodnot. Další implementace *AxisCategoryAndCondition*, *AxisCategoryOrCondition* a *AxisCategoryNot-Condition* představují množinové operace průnik, sjednocení a doplněk, pomocí kterých lze množiny definující data skládat. Poslední implementací je *AxisCategoryAllCondition*, která představuje množinu všech dat.

Kromě balíčku *chart* byl doimplementován velký počet tříd do balíčku *gui*. Tyto třídy se starají o grafické uživatelské rozhraní a vykreslení vizualizace. Vykreslení vizualizace grafu paralelních množin zajišťuje třída *ParallelSetsChartPanel*, která rozšiřuje původní třídu *ChartPanel*, stejně jako třída *ParallelCoordinatesChartPanel*. Pro grafické uživatelské rozhraní byly přidány třídy obsluhující jednotlivá menu a dialogy pro vytváření vlastní osy, editaci kategorií a nastavení zobrazení grafu.

Dále byla implementována třída *ParallelSetsCreationThread* z balíčku *workerThreads*, která reprezentuje vlákno pro vytvoření grafu paralelních množin.



Obrázek 34: Diagram tříd balíčku chart – šedě označené třídy jsou původní třídy aplikace XDat, bílé jsou nově přidané třídy, zeleně označený je interface a modře abstraktní třídy

5 Výsledky

V rámci této práce byla navržena a implementována aplikace, která umožňuje vizualizaci n-rozměrných heterogenních dat metodou paralelních množin a nově navrženou metodou Set Rivers. V této kapitole jsou na třech různých souborech dat prezentovány výsledky v podobě ukázky vizualizací, kterých lze pomocí navržené aplikace dosáhnout. V následující kapitole je popsáno uživatelské testování, které bylo provedeno za účelem ověření funkčnosti aplikace a srovnání vizualizačních metod.

5.1 Vstupní data

K prezentování výsledků implementované aplikace byla použita následující reálná data:

A. Titanic

První soubor dat obsahuje informace o cestujících Titaniku. Jedná se o kategorická data, která obsahují 2201 položek. Atributy dat jsou třída cestujících, věk (dospělý / dítě), pohlaví a informace, zda cestující přežil nehodu. Tento soubor dat obsahuje pouze čtyři dimenze, takže při jeho vizualizaci by neměl nastat problém ani u metody paralelních množin.

B. Cars

Dalším souborem dat jsou informace o automobilech. Tato data obsahují 391 položek a 9 atributů. Atributy jsou parametry automobilů, jako například počet válců, výkon nebo rychlost, a nabývají spojitých i kategorických hodnot.

C. Mushrooms

Poslední soubor dat, který byl použit pro prezentaci výsledků, jsou data o houbách. Každá datová položka odpovídá jednomu typu houby. Atributy těchto dat reprezentují parametry hub, například velikost, tvar nebo barvu klobouku. Celkový počet atributů je 23 a data obsahují 8142 položek. Díky velkému počtu dimenzí lze na těchto datech prověřit metodu Set Rivers v oblasti, kde metoda paralelních množin nefunguje příliš dobře.

5.2 Metoda paralelních množin

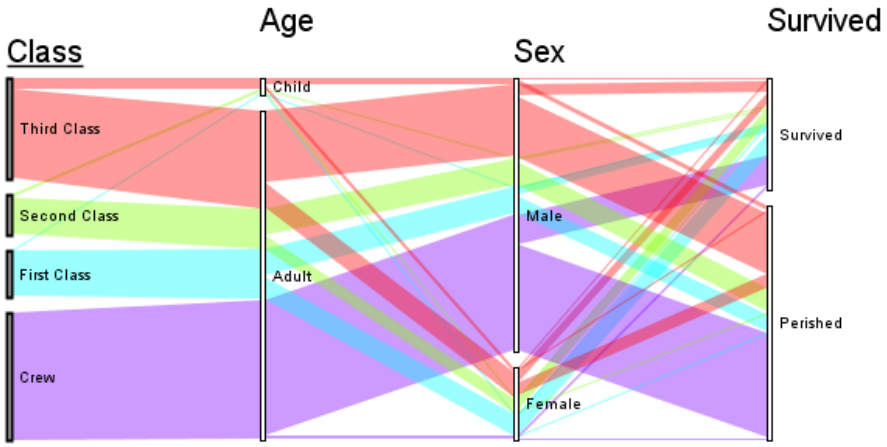
Základní vizualizací, kterou lze pomocí navržené aplikace vytvořit, je vizualizace paralelních množin. Následují ukázky této vizualizace se stromovým režimem rozložení rovnoběžníků, poté

se svazkovým režimem, a poté ukázka vlastní osy vytvořené uživatelem. Všechny ukázky jsou předvedeny na třech výše uvedených souborech dat, kde u datového souboru *A* (Titanic) jsou zobrazeny osy pro všechny dimenze, u souboru *B* (Cars) je zobrazeno sedm dimenzí a u souboru *C* (Mushrooms) je zobrazeno jedenáct dimenzí. V případě vizualizace s vlastní osou jsou skryty dimenze, které jsou obsaženy ve vlastní ose.

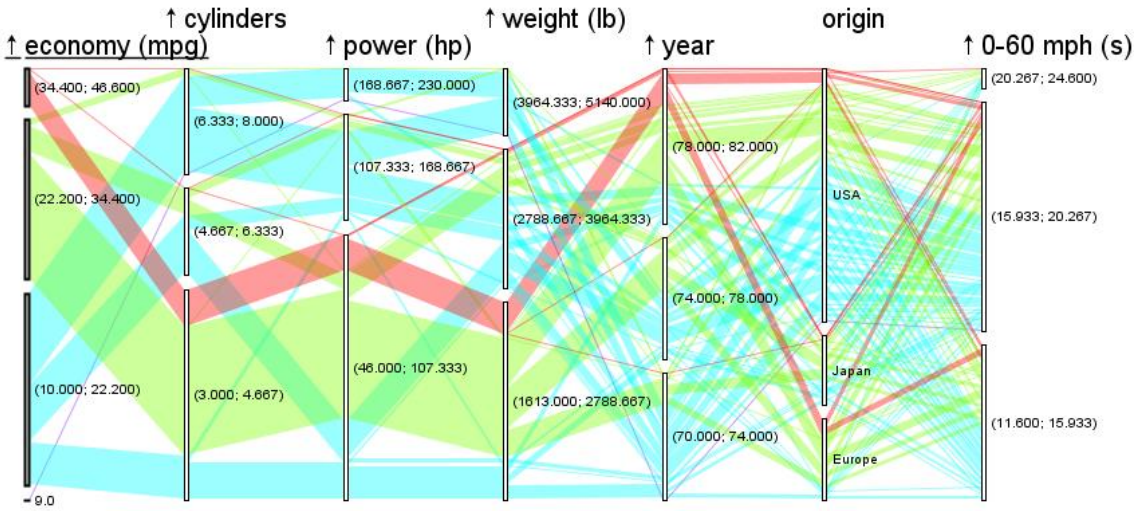
5.2.1 Stromové rozložení

Vizualizace stromového rozložení je ukázána na následujících obrázcích tak, že Obrázek 35a znázorňuje vizualizaci na datovém souboru *A*, Obrázek 35b na souboru *B* a Obrázek 35c na souboru *C*. Tento způsob číslování obrázků platí i u všech následujících vizualizací v této kapitole.

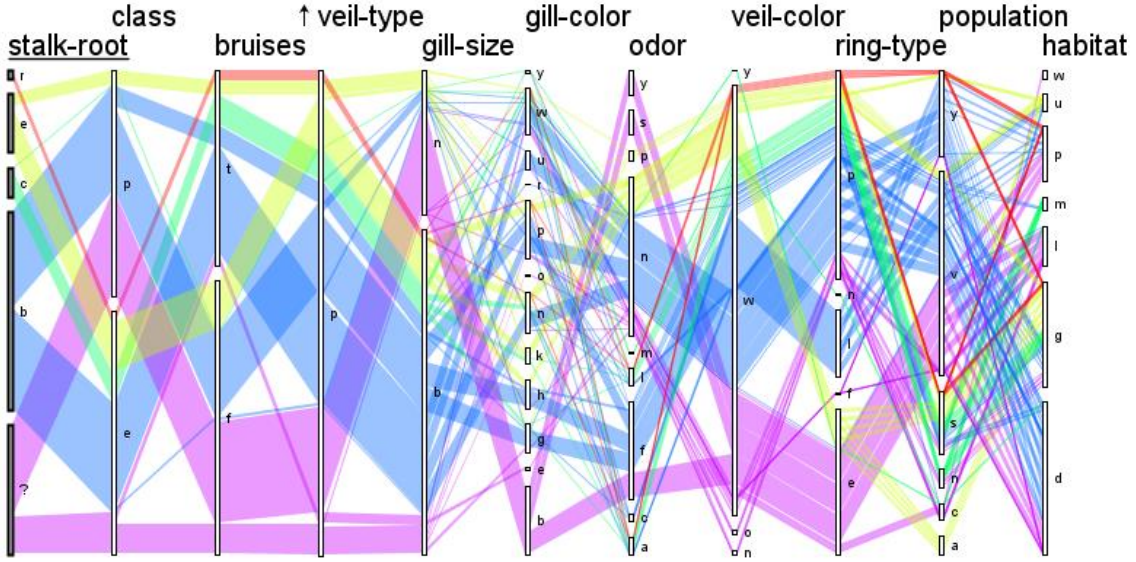
Výsledek vizualizace stromového rozložení odpovídá popisu metody z kapitoly 2.6. První případ se čtyřmi osami (Obrázek 35a) je přehledný a dobře čitelný, ale ve druhém a třetím (Obrázek 35b, c) se projevuje nedostatek stromového rozložení paralelních množin, kterým je špatná přehlednost grafu s více osami. Lze sledovat, že ve druhém a třetím případě se u páté až šesté osy stává vizualizace nepřehlednou.



(a)



(b)

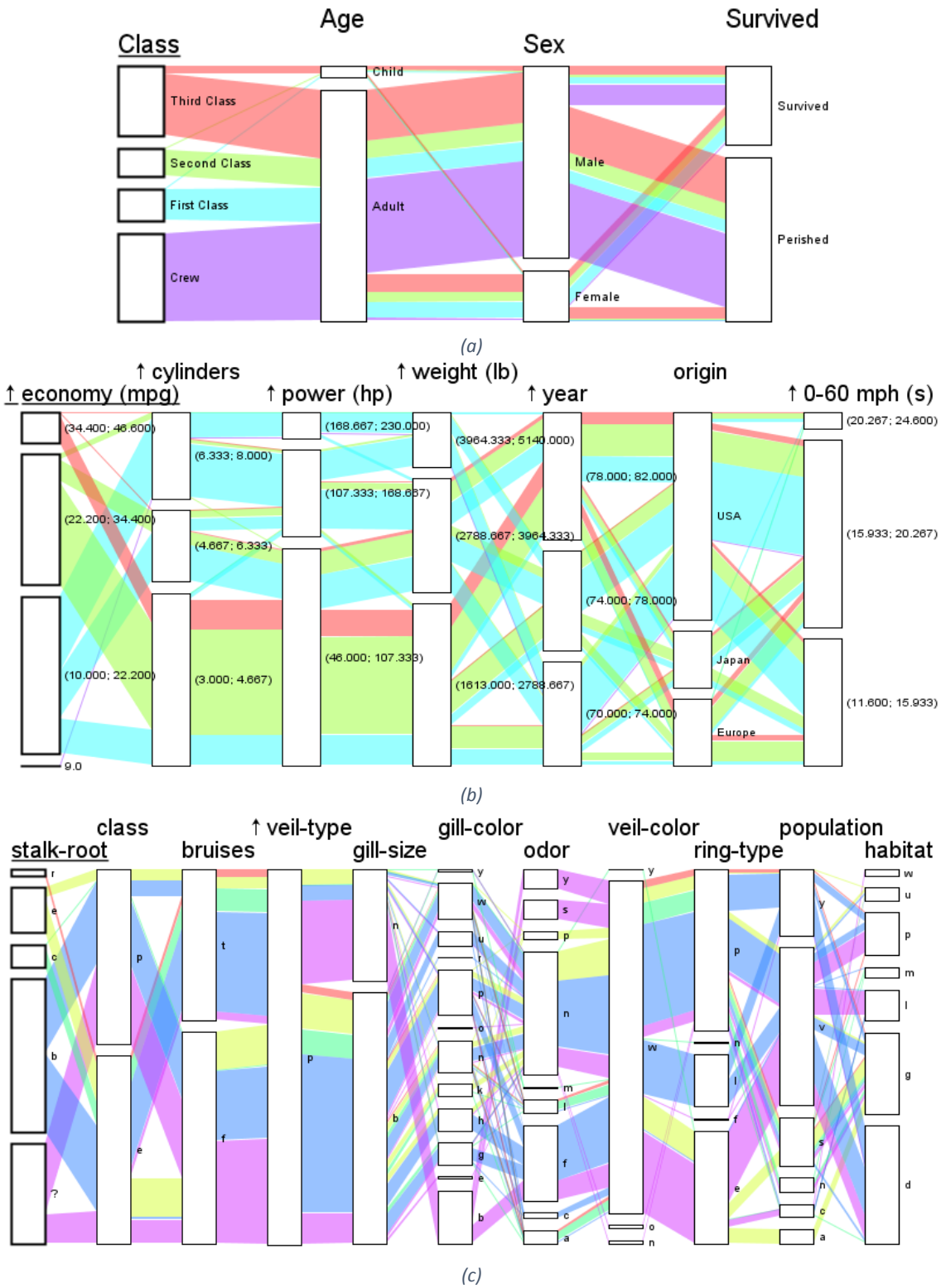


(c)

Obrázek 35: Vizualizace stromového rozložení paralelních množin

5.2.2 Svazkové rozložení

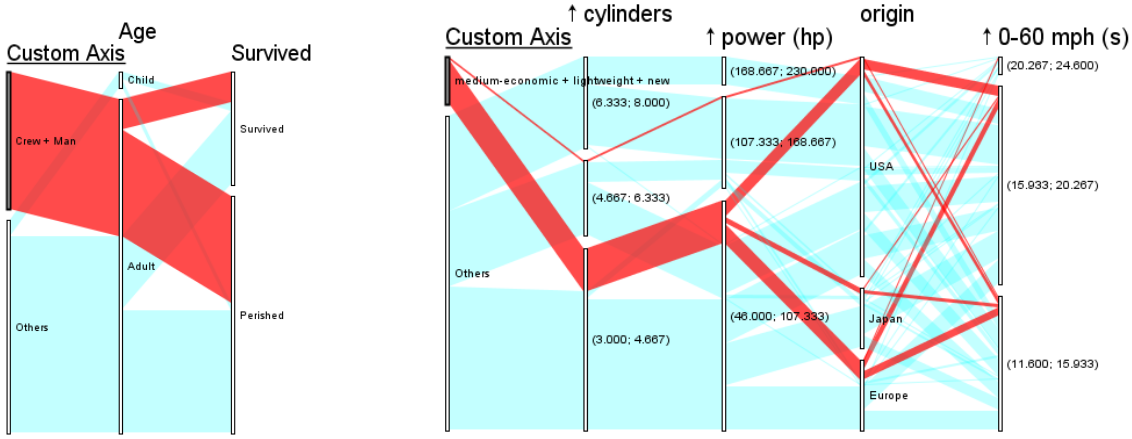
Výsledná vizualizace svazkového rozložení také odpovídá popisu této metody a lze sledovat, že pro větší počet os (Obrázek 36c, d) je vizualizace přehlednější než v případě stromového rozložení. Projevuje se zde ale výše popsaný nedostatek svazkového rozložení, kterým je obtížnější sledování závislostí mezi více osami.



Obrázek 36: Vizualizace svazkového rozložení paralelních množin

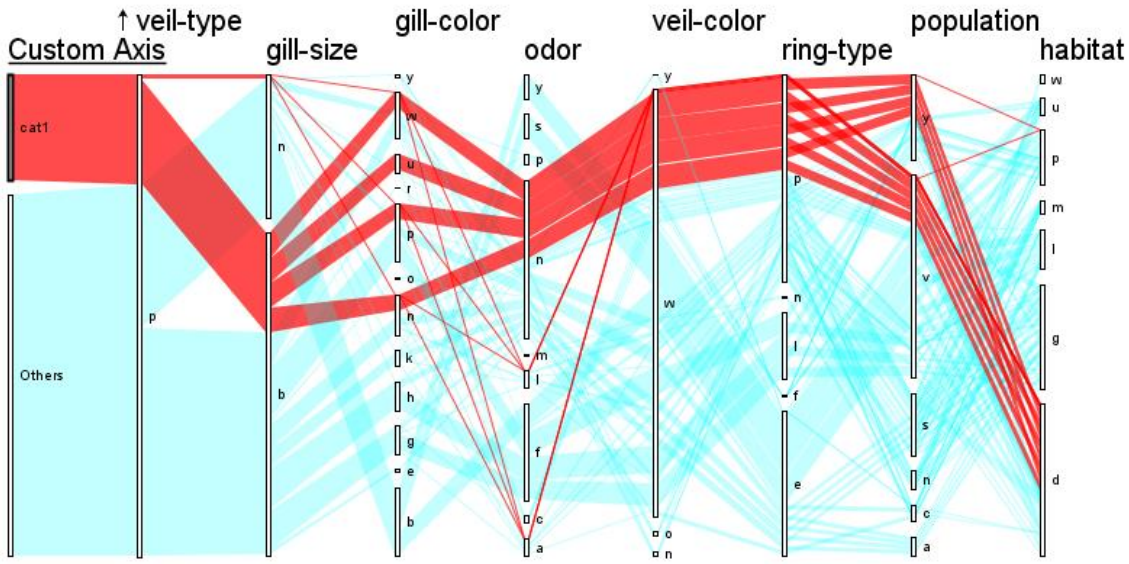
5.2.3 Vlastní osa

Při stromovém rozložení aplikace umožňuje uživateli vytvořit vlastní osu, pomocí které může skládat dimenze. Tím lze omezit počet os a graf paralelních množin je potom přehlednější. Výsledek je ukázán opět pro tři typy dat (Obrázek 37a, b, c), kde vlastní osa s názvem *Custom Axis* je vždy první osa ve vizualizaci.



(a)

(b)

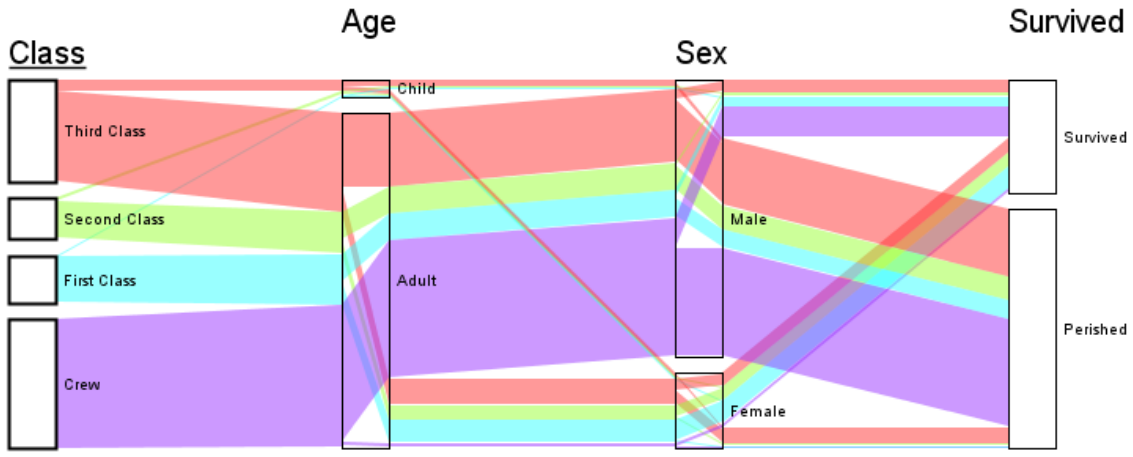


(c)

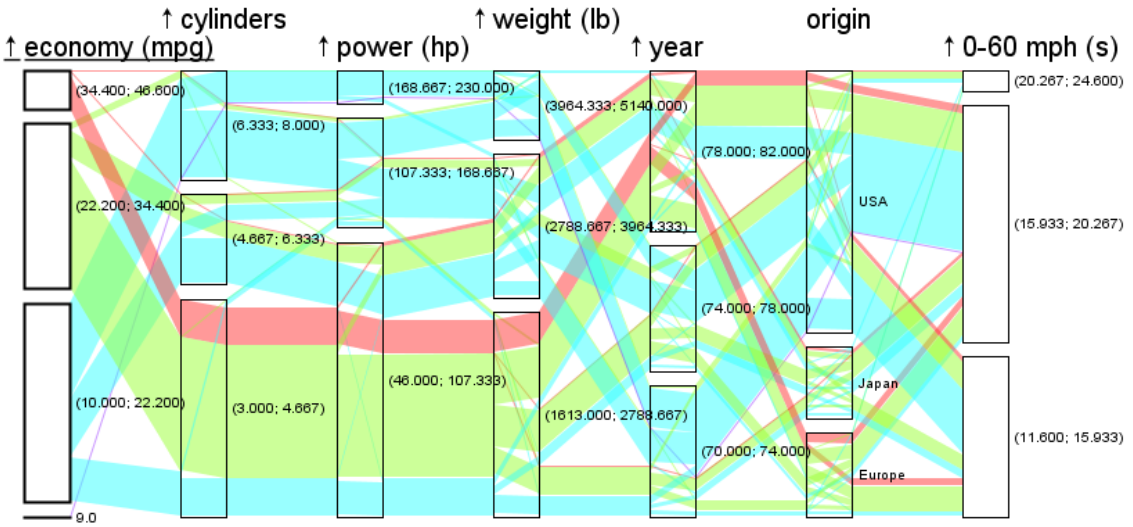
Obrázek 37: Vizualizace stromového rozložení s vlastní osou

5.3 Metoda Set Rivers

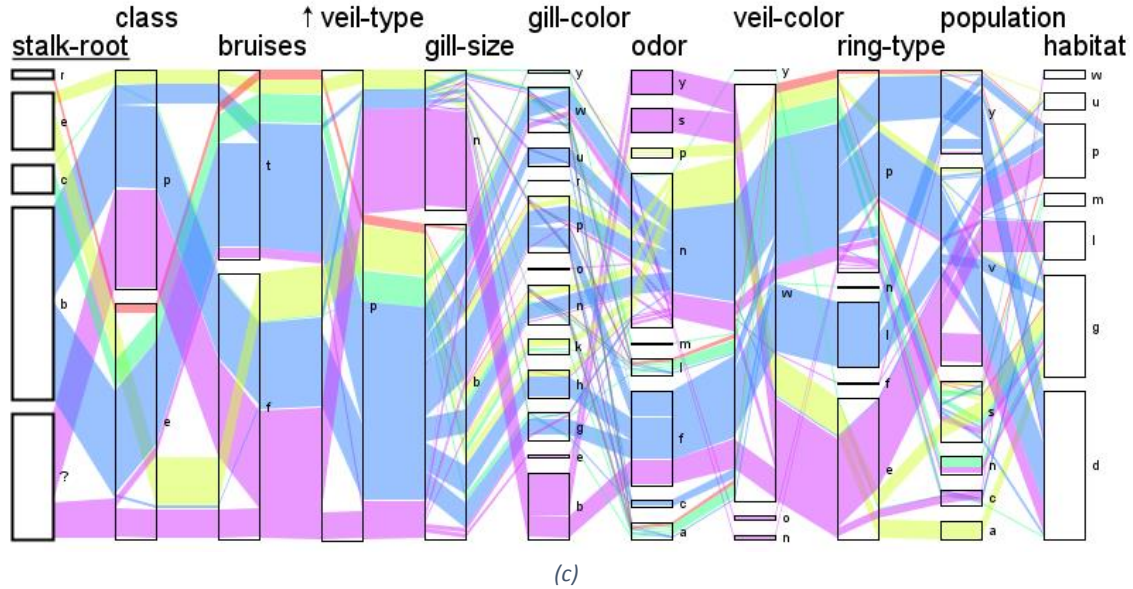
Navržená aplikace dále umožňuje vytvořit vizualizaci metodou Set Rivers, která byla popsána v kapitole 3.1.2. Výsledek této vizualizace je opět ukázán na třech typech dat (Obrázek 38a, b, c).



(a)



(b)

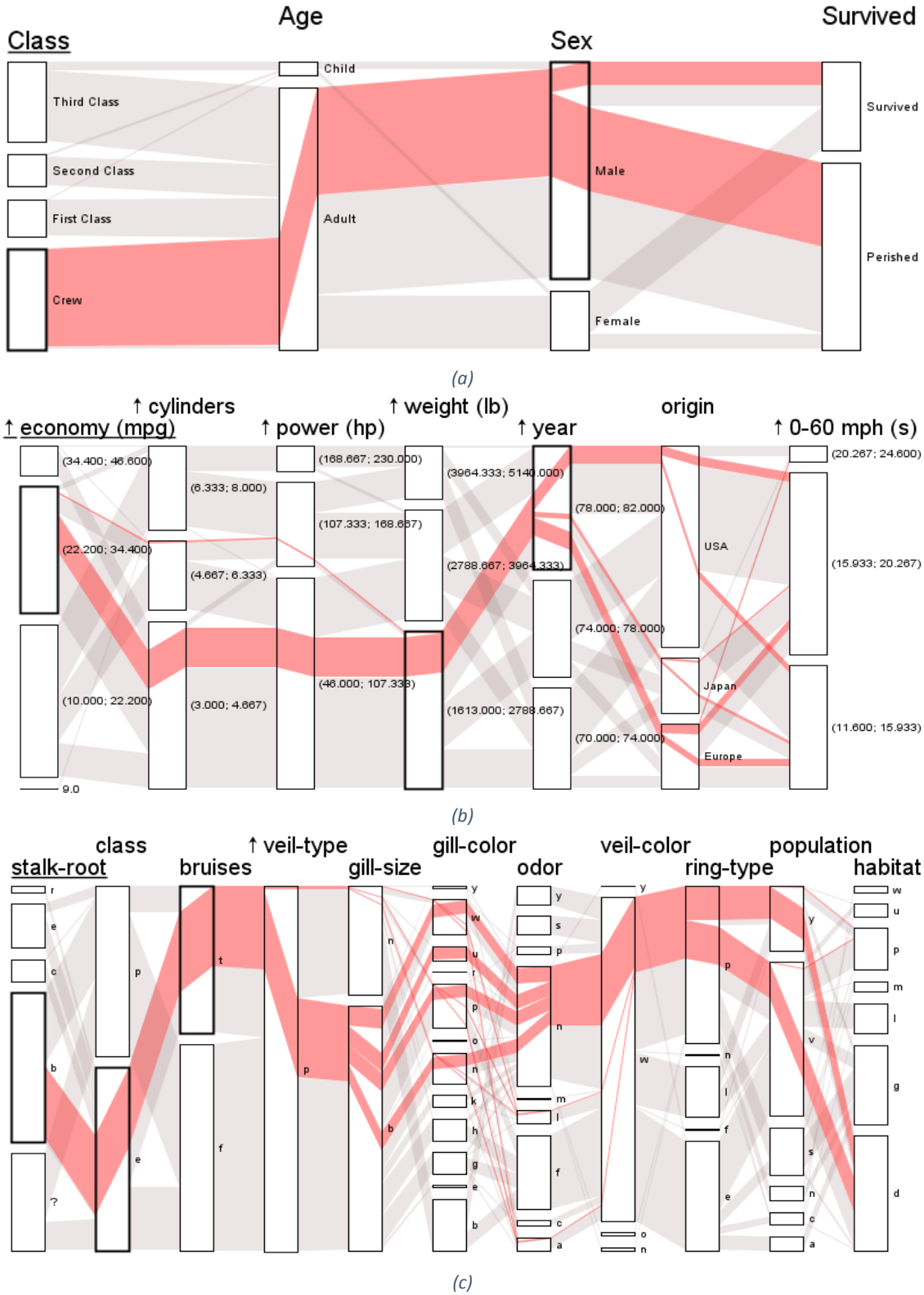


(c)

Obrázek 38: Vizualizace metodou Set Rivers

5.3.1 Jeden filtr

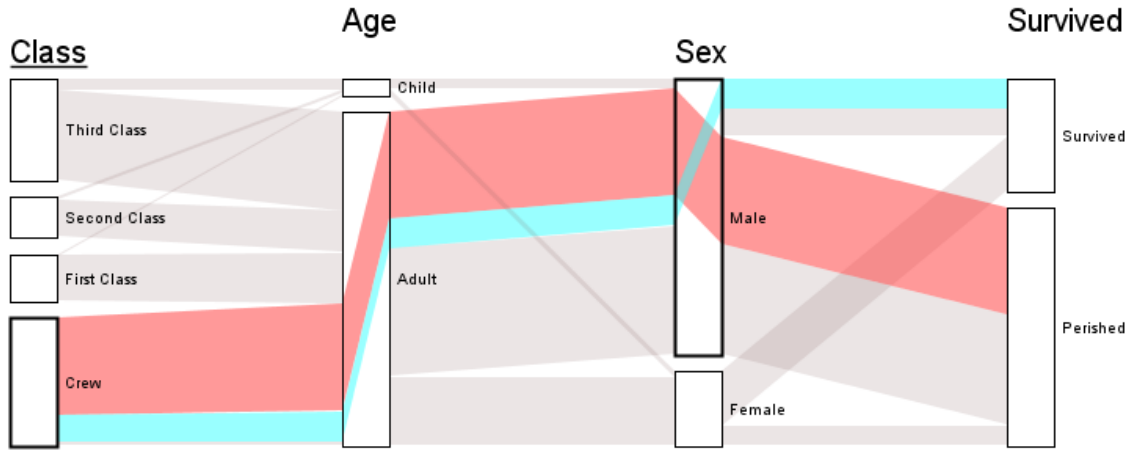
V rámci metody Set Rivers aplikace umožňuje filtrovat data vybíráním kategorií, kterými mají data procházet. To vede k podrobnějšímu zobrazení jen takových dat, které jsou v oblasti zájmu uživatele (Obrázek 39a, b, c).



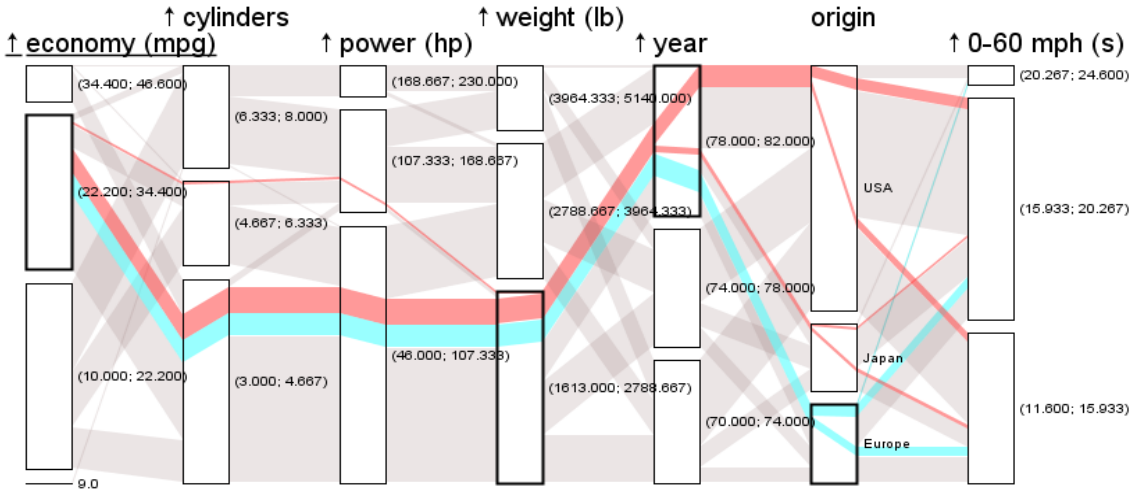
Obrázek 39: Filtrování v rámci metody Set Rivers

5.3.2 Více filtrů

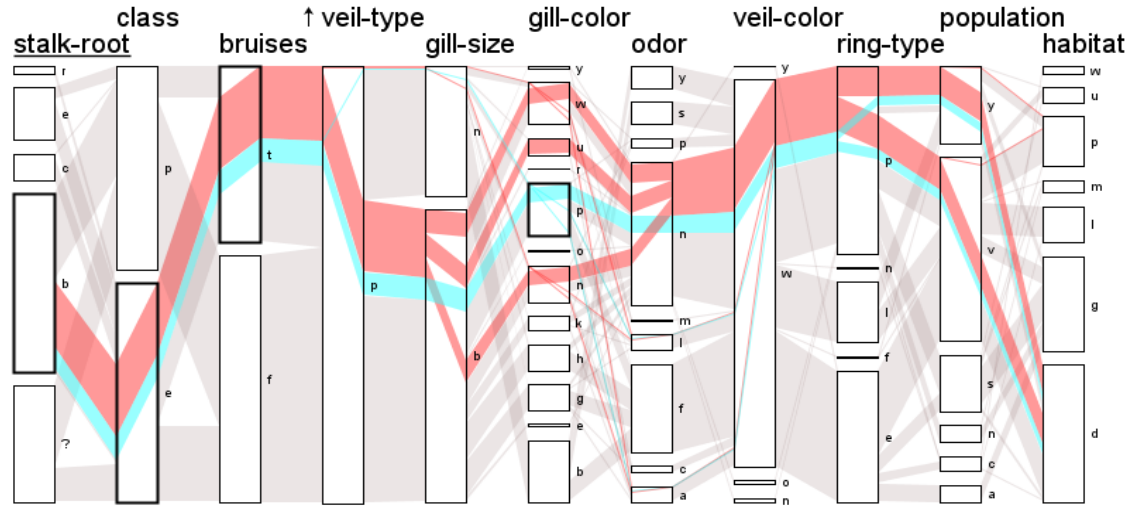
Filtrů může být zobrazeno i více zároveň (Obrázek 40a, b, c). Tím lze oddělit a porovnat různé skupiny dat. Například v prvním případě (Obrázek 40a) jsou modře vyznačeni muži z posádky, kteří byli zachráněni, a červeně ti, co nebyli.



(a)



(b)

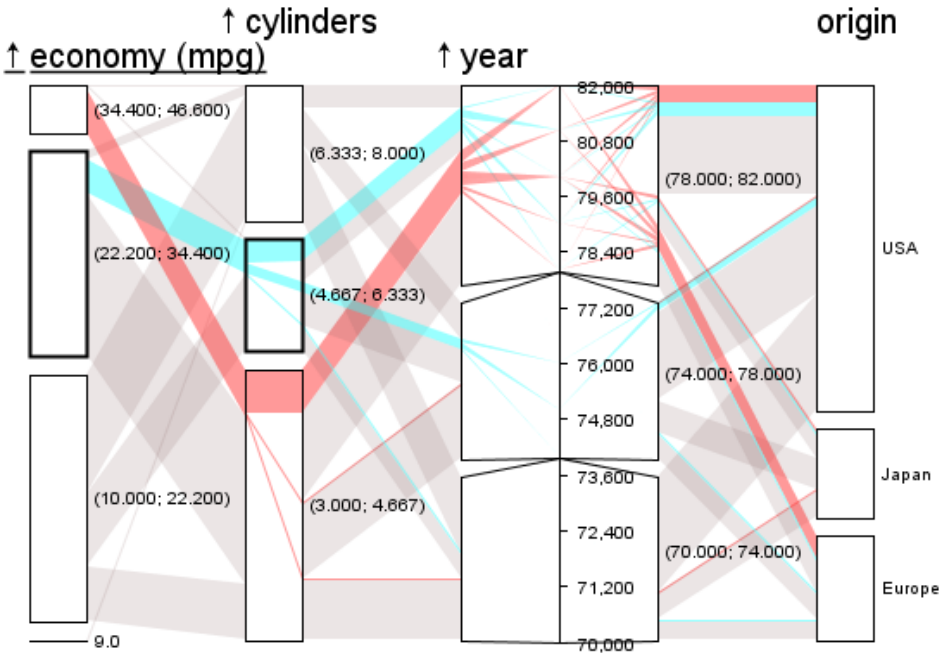


(c)

Obrázek 40: Zobrazení více filtrů v rámci metody Set Rivers

5.3.3 Číselná osa

Dále aplikace umožňuje zobrazení číselné osy uvnitř osy kategorické, tak, jak bylo popsáno v návrhu (kapitola 3.4.13). Díky tomu lze podrobněji sledovat hodnoty dat na osách spojitých atributů. Výsledek tohoto zobrazení je na následujícím obrázku (Obrázek 41) na příkladu časové osy *year*.



Obrázek 41: Vizualizace se zobrazením číselné osy

6 Uživatelské testování

Implementovaná aplikace byla otestována uživatelskými testy s cílem ověřit funkčnost nově navržené metody Set Rivers a porovnat práci s touto metodou a klasickou metodou paralelních množin.

6.1 Popis testovací metody a metod pro zpracování výsledků

Jako testovací metoda byl zvolen kvalitativní test použitelnosti s uživateli [24]. Při této metodě je vybráno několik uživatelů, kteří patří do cílové skupiny, a ti jsou pozváni k testu. Test probíhá tak, že každý participant samostatně plní předem připravené úkoly v testovaném systému. Celý průběh plnění úkolů je monitorován, je zaznamenáváno, kde dochází k případným problémům, a je měřen čas potřebný ke splnění úkolu. Po dokončení plnění úkolů uživatel vyplní dotazník, jehož součástí může být i otevřená otázka s neformální zpětnou vazbou. Následně jsou na základě záznamů z testů popsány problémy, které se při testování vyskytly a je vyhodnocen výsledek testu. Touto metodou lze důkladně prověřit použitelnost testovaného systému nebo metody a lze objevit a popsat problémy, které nebyly zatím zjištěny.

Dotazník, který uživatelé vyplnili po skončení testu, byl sestaven tak, že obsahoval tři uzavřené otázky a jednu otevřenou. Uzavřené otázky měly varianty odpovědí dle Likertovi škály [14] a byly zaměřeny na subjektivní hodnocení testované metody. Otevřená otázka poskytovala uživatelům možnost vyjádřit zpětnou vazbu, na kterou nebyl během testování prostor.

Odpovědi z dotazníku byly číselně ohodnoceny tak, že Likertově škále odpovídali hodnoty 1 až 5, a poté byly společně s časy plnění úkolů statisticky zpracovány.

Pro každou otázku a testovanou vizualizační techniku byl spočítán aritmetický průměr odpovědí. Pro vizualizační techniku *T* a otázku *Q* byl průměr odpovědí spočítán jako:

$$P = \frac{1}{N}\sum_{i=1}^{N} x_i$$

kde x_i je číselná hodnota odpovědi uživatele *i* na otázku *Q* a *N* je počet uživatelů, kteří testovali techniku *T*.

Dále byl pro každou takovou hodnotu spočítán 95% interval spolehlivosti. Pomocí tohoto intervalu lze na základě dat naměřených na vzorku uživatelů říci, kde by měla s pravděpodobností 0,95 ležet skutečná hodnota [17].

Pro časy plnění úkolů byl také spočítán průměr a interval spolehlivosti pro každou testovanou techniku. Před výpočtem intervalu spolehlivosti u časových dat byly ale hodnoty logaritmicky transformovány. Tato transformace byla provedena proto, že pro přesný výpočet intervalu spolehlivosti jsou vyžadována data, která mají přibližně normální rozdělení, ale časová data nasbíraná při uživatelském testování normálnímu rozdělení neodpovídají. Aby bylo dosaženo přibližně normálního rozdělení, je třeba data transformovat. Pro transformaci časových dat je dle dostupných pramenů [9] nejvhodnější právě logaritmická transformace. Kvůli vlastnostem rozdělení časových dat je také vhodnější reportovat geometrický průměr místo aritmetického. Logaritmus geometrického průměru kladných hodnot je totiž roven aritmetickému průměru logaritmů:

$$\ln G = \frac{1}{N}\sum_{i=1}^{N} \ln x_i$$

Geometrický průměr tedy odpovídá aritmetickému na logaritmické transformaci.

6.2 Průběh tesu

Pro testování bylo vybráno 10 lidí, kteří mají běžnou znalost práce s počítačem a alespoň základní znalost práce s grafy a vizualizacemi. Mezi vybranými uživateli byli zastoupeni muži i ženy, všichni spadali do věkové kategorie 18 až 30 lety. Tito uživatelé byli rozděleni do dvou skupin, kde jedna skupina testovala metodu paralelních množin a druhá metodu Set Rivers. Pro obě skupiny byl shodný průběh testu, testovací úkoly i testovací data. Lišila se pouze metoda, kterou uživatelé k plnění úkolů používali.

Na začátku testu byli participanti seznámeni s průběhem testu a principy uživatelského testování. Byli informováni o tom, že budou sbírána data z plnění úkolů a společně s daty z dotazníku budou uchovávána pod anonymním identifikátorem. Dále byli požádáni o souhlas se zpracováváním poskytnutých údajů a byli informování, že mohou kdykoliv svůj souhlas odvolat. Poté následovaly dvě fáze testu. První fáze byla tréninková, s cílem seznámit participanta s testovaným nástrojem a vizualizační metodou, druhá byla testovací, z níž se sbírala data o plnění úkolů.

Na počátku tréninkové fáze byly participantovi vysvětleny principy vizualizační metody, která mu byla přidělena, a poté principy práce s navrženou aplikací. Práce s aplikací byla moderátorem testu názorně předvedena. Poté byl participantovi prezentován tréninkový úkol (viz příloha A) a byl požádán o jeho splnění na tréninkových datech. Měl-li participant při plnění úkolu problémy, moderátor mu pomohl a problematiku znovu vysvětlil, tak, aby byl participant schopen úkol zopakovat sám. Když byl moderátor přesvědčen, že participant chápe principy vizualizační metody a samostatně zvládá práci s aplikací, byla tréninková fáze ukončena a pokračovala fáze testovací.

V testovací fázi byl participantovi prezentován testovací úkol a aplikace byla nastavena do počátečního stavu pro test. Počáteční stav obsahuje pro jednu skupinu participantů vizualizaci testovacích dat metodou paralelních množin s nastaveným stromovým rozložením a pro druhou metodou Set Rivers. Testovací úkol i data se lišila od tréninkových, ale byla založena na stejných principech práce. Při plnění úkolu v testovací fázi participant postupoval samostatně a moderátor si poznamenával průběh plnění úkolu, případné problémy a čas, za který je participant schopen úlohu dokončit. Po splnění úkolu byl participantovi předán dotazník (viz příloha B), kde vyplnil otázky týkající se subjektivního hodnocení testované metody.

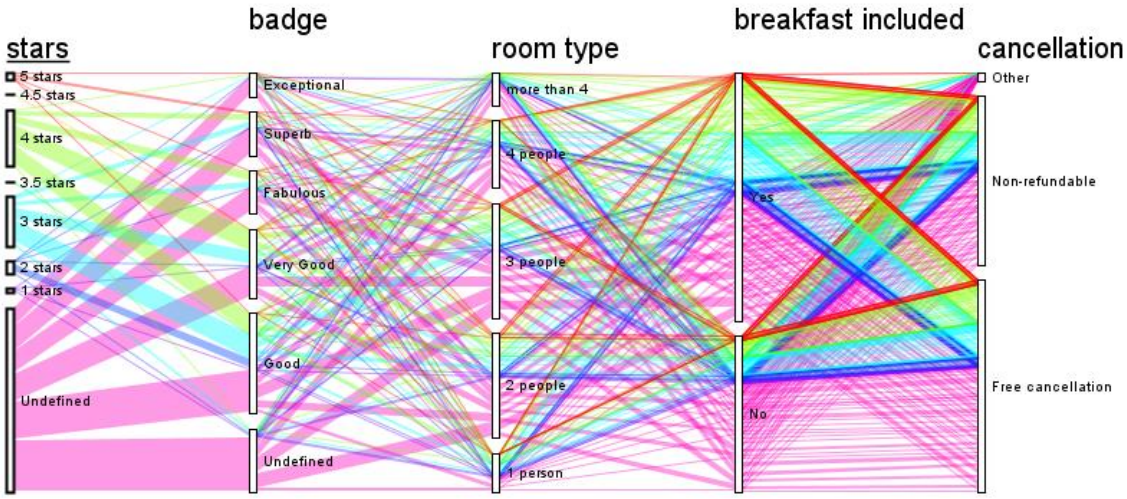
Testování probíhalo v laboratoři U-LAB na katedře počítačové grafiky ČVUT, kde měl participant k dispozici stolní počítač v běžné konfiguraci – monitor, klávesnice, myš. Rozlišení monitoru, který byl při testování použit, je 1920 x 1080.

Proces testování jednoho člověka trval v průměru přibližně hodinu, kdy 35 minut zabrala tréninková fáze a 25 fáze testovací.

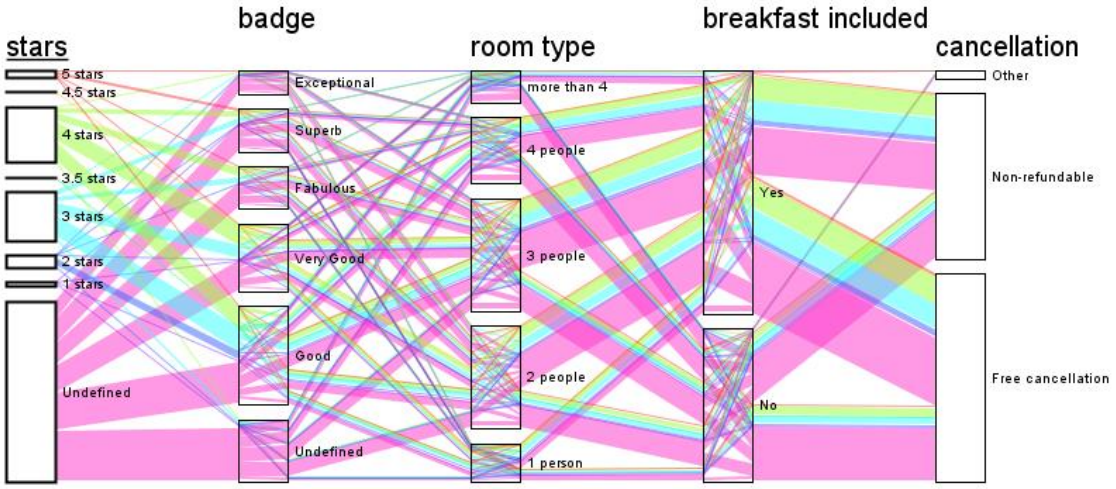
6.3 Testovací data

Pro testování byly použity dva soubory dat. Jeden soubor byl použit pro tréninkovou fázi, kde se uživatel seznamoval s aplikací a metodou, na druhém plnil úkol v testovací fázi. Oba soubory dat představují informace o hotelových pokojích a charakter dat je také v obou případech stejný. Jedná se o heterogenní n-rozměrná data, která v některých dimenzích nabývají spojitých číselných hodnot (například cena za noc), v některých kategorických (například hodnocení hotelu) a někde nabývají jen jedné ze dvou hodnot ANO / NE (například snídaně v ceně).

Pro tréninkovou fázi byla použita data o hotelech v Římě. Tyto data obsahují 14 379 položek. Participantovi byla prezentována předem sestavená vizualizace těchto dat, ve které bylo zobrazeno pět atributů: počet hvězdiček hotelu, hodnocení, počet lůžek, snídaně v ceně a možnosti zrušení rezervace. Počáteční stav této vizualizace při začátku testu je na následujícím obrázku (Obrázek 42).



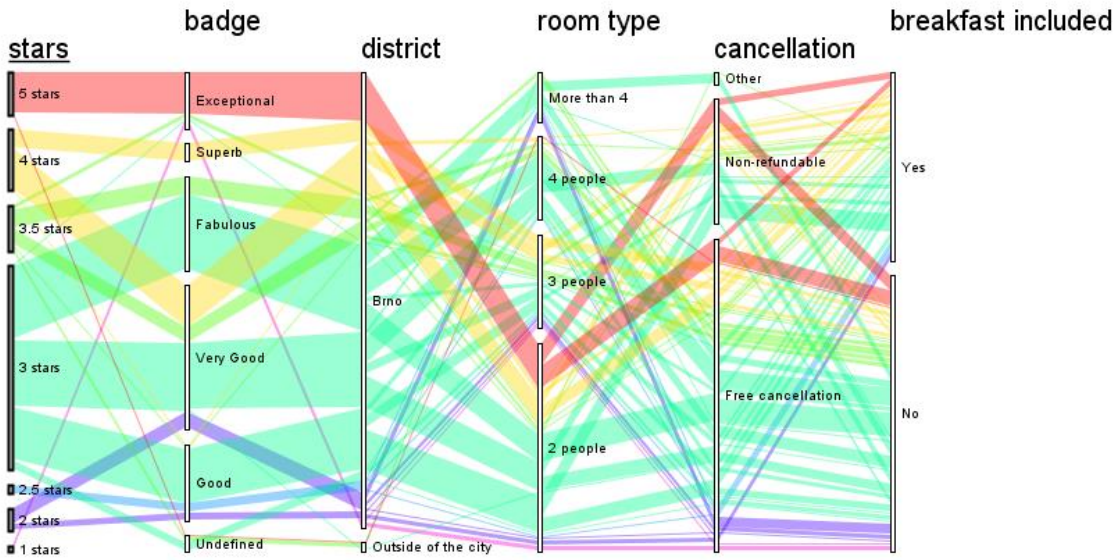
(a)



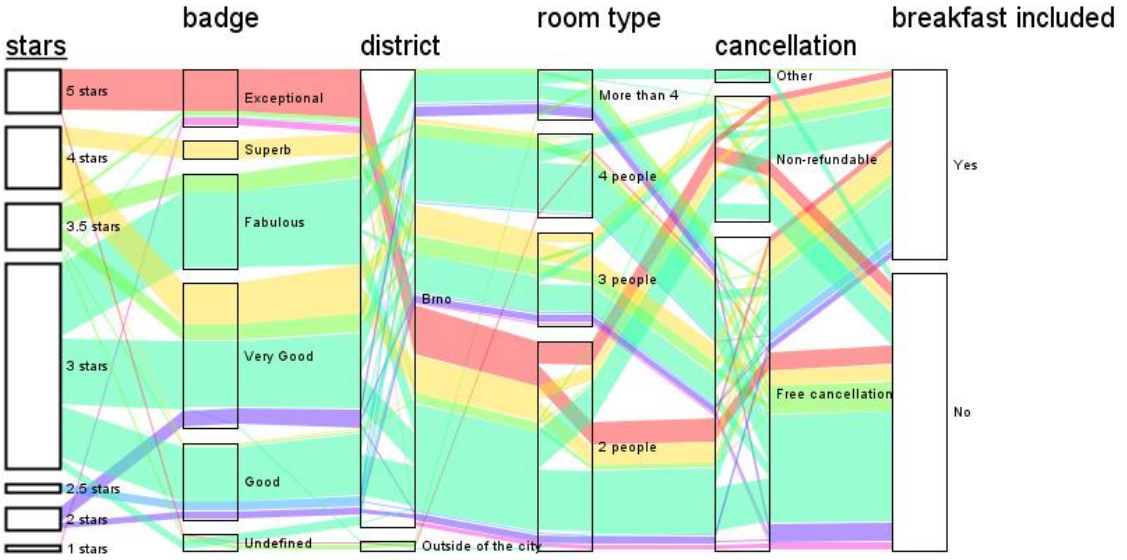
(b)

Obrázek 42: Počáteční stav vizualizace pro tréninkovou fázi testu - (a) metoda paralelních množin, (b) metoda Set Rivers

Pro testovací fázi byla použita data o hotelech v Brně, která obsahují méně položek (172), ale mají stejný charakter. Předem sestavená vizualizace obsahovala šest atributů, kde pět atributů bylo shodných jako u tréninkových dat a šestým byla oblast hotelu. Tato vizualizace, tak, jak byla prezentována participantům při začátku testu, je znázorněna na následujícím obrázku (Obrázek 43).



(a)



(b)

Obrázek 43: Počáteční stav vizualizace pro testovací fázi - (a) metoda paralelních množin, (b) metoda Set Rivers

6.4 Testovací úkoly

Před prezentováním testovacích úkolů byl participantovi nastíněn scénář případu použití aplikace. Poté byly prezentovány úkoly, pro tréninkovou a testovací fázi. Očekávaným výsledkem obou těchto úkolů byl co nejpřesnější popis závislostí mezi co největším počtem zobrazených atributů. Počet atributů, mezi nimiž je participant schopen popsat závislosti, a přesnost tohoto popisu, bylo klíčovými kritérii pro vyhodnocení úspěšnosti splnění úkolu.

Scénář testovacího případu použití:

Jste analytik / analytička cestovní kanceláře a vaším úkolem je analyzovat data o pokojích ve všech nabízených hotelech v Římě.

Úkol pro tréninkovou fázi:

Najděte všechny dvoulůžkové pokoje v hotelech bez hvězdičky (stars = undefined). Můžete popsat závislosti mezi jednotlivými atributy?

Úkol pro testovací fázi:

Najděte všechny čtyřlůžkové pokoje. Jaké jsou relace / závislosti mezi atributy pro tyto pokoje?

6.5 Nálezy

V této části jsou popsány problémy, které se projevily během uživatelského testování. Nálezy jsou rozděleny podle testovaných metod.

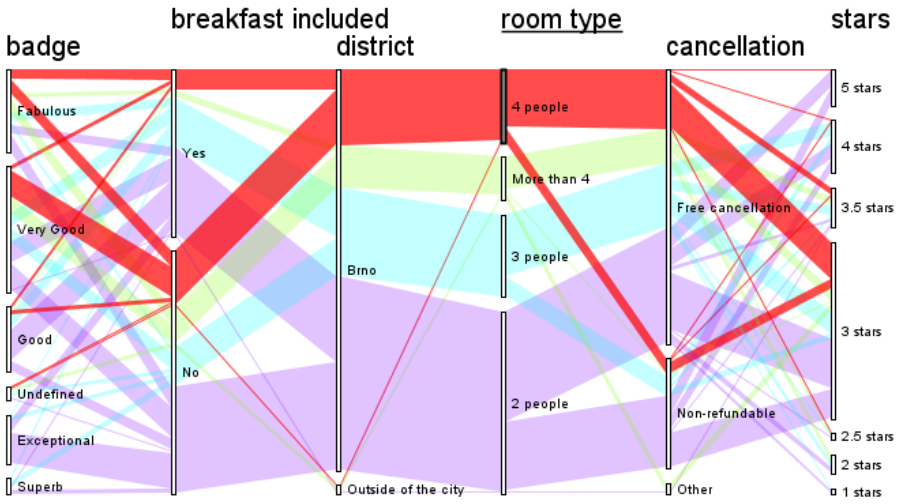
6.5.1 Metoda 1 – paralelní množiny

1) Problémy s popisem závislostí mezi vzdálenějšími osami

Všem participantům se bez problému dařilo popsat závislosti mezi prvními třemi osami. U čtvrté osy si někteří nebyli jistí a u páté a šesté popisovala většina participantů závislosti nepřesně a dopouštěli se chybných závěrů.

2) Problém aktivní osy uprostřed grafu se stromovým rozložením

Při snaze dosáhnout co nejpřehlednějšího zobrazení přesunul jeden z participantů aktivní kořenovou osu doprostřed grafu. Tímto způsobem ale požadovaného výsledku nedosáhl, protože pak nebyl schopen popsat závislosti mezi levou a pravou částí grafu.



Obrázek 44: Problém kořenové osy uprostřed grafu

3) Zdlouhavé a uživatelsky nekomfortní vytváření vlastní osy

Jeden z participantů řešil popis složitějších závislostí vytvářením několika vlastních os. Díky tomu byl schopen závislosti popsat. Vytváření vlastních os bylo ale natolik zdlouhavé a nekomfortní, že po chvíli tento přístup opustil.

6.5.2 Metoda 2 – Set Rivers

1) Problém práce s filtry

Participanti dokázali vyčíst závislosti mezi prvními třemi osami bez přidání filtrů, pro popis dalších os se snažili používat filtry. Používání jednoho filtru způsobem, že do něj interaktivně přidávali a odebírali kategorie a sledovali, jak se vybraná množina dat mění, nedělalo nikomu z participantů problém. Pro složitější závislosti bylo ale třeba vytvořit více filtrů a rozdělit s nimi nejednoznačné svazky. To se dvěma z pěti participantů v některých případech nepodařilo. Tři zbývající participanti dokázali svazky rozdělit napříč celým grafem a vyčíst potřebné informace.

2) Přebarvování filtrů při přidání nového filtru

Jednomu participantovi přišlo matoucí, že při přidání nového filtru dojde ke změně barvy stávajících filtrů.

Poznámka: Toto je způsobeno snahou generovat co nejodlišnější barvy pro všechny filtry. Proto když se změní počet filtrů, změní se i všechny barvy. Z hlediska konzistence pro uživatele by ale nejspíš bylo lepší řešení barvy stávajících filtrů ponechat na úkor menšího rozdílu barevného odstínu nového filtru.

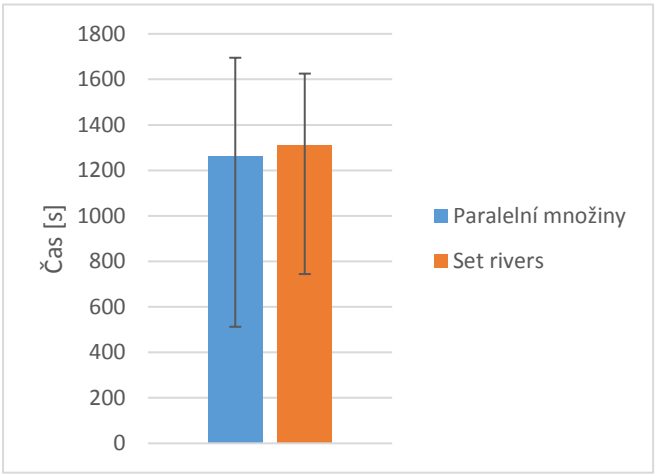
6.6 Statistické zpracování výsledků

Během testu byl měřen čas, za který participant vyřešil testovací úlohu. Tyto časy byly poté zpracovány pro účel porovnání rychlosti řešení úlohy pomocí metody paralelních množin a Set Rivers. Naměřené časy jsou uvedeny v následující tabulce (Tabulka 1).

Tabulka 1: Časy řešení testovací úlohy

Metoda paralelních množin		**Metoda Set Rivers**	
Participant č.	**Délka testu [s]**	**Participant č.**	**Délka testu [s]**
1	1430	2	1093
3	1267	4	1079
6	1433	5	1510
8	840	7	1452
10	1481	9	1500

Geometrický průměr časů řešení úlohy pomocí metody paralelních množin byl dle naměřených dat vypočten jako 1 264,27 sekund. Při řešení této úlohu pomocí metody Set Rivers, vyšel tento průměrný čas 1 311,40 sekund. 95% interval spolehlivosti pro tuto veličinu je (943,01; 1694,96) pro metodu paralelních množin a (1058,10; 1625,34) pro metodu Set Rivers. V následujícím grafu (Obrázek 45) jsou tyto hodnoty znázorněny. Průměrný čas potřebný pro řešení úlohy metodou paralelních množin je znázorněn modrým sloupcem a pro metodu Set Rivers oranžovým sloupcem. Chybové úsečky znázorňují intervaly spolehlivosti.



Obrázek 45: Graf času řešení testovací úlohy

Z grafu je patrné, že průměrné hodnoty časů dvou testovaných metod se od sebe příliš neliší. Jejich intervaly spolehlivosti jsou ale poměrně velké a extrémně se překrývají, takže z těchto dat není možné stanovit závěr, že by některá z metod byla rychlejší než druhá. Nelze ani říci, že jsou metody stejně rychlé, protože podobnost průměrných časů může být způsobena malým počtem vzorků dat.

Dále byly zpracovány odpovědi na otázky, které participanti zodpověděli v rámci post-test dotazníku. Dotazník obsahoval tři následující otázky týkající se subjektivního hodnocení testovaných metod:

- Otázka1: Metoda pro mě byla komfortní.
- Otázka2: Metoda mi dovolila pracovat rychle.
- Otázka3: V plnění úkolů jsem si byl jistý.

Odpovědi na otázky měli formu Likertovi škály, kde participanti u každé otázky vybrali jednu z možností: rozhodně souhlasím, souhlasím, nevím, nesouhlasím nebo rozhodně nesouhlasím. Tyto odpovědi byly ohodnoceny čísly 1 až 5, kde 1 reprezentovala odpověď rozhodně souhlasím a 5 rozhodně nesouhlasím.

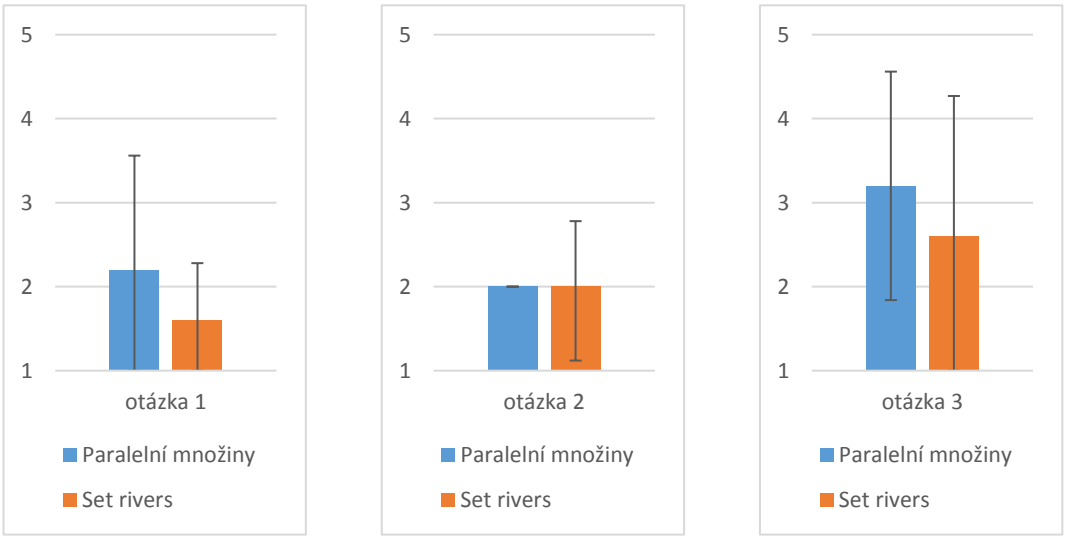
Z těchto odpovědí byl spočítán aritmetický průměr pro danou otázku a metodu a interval spolehlivosti. Vypočítané hodnoty jsou uvedeny v následující tabulce (Tabulka 2).

Tabulka 2: Průměrné odpovědi na otázky ohledně hodnocení testovaných vizualizačních metod

		Otázka 1	Otázka 2	Otázka 3
Metoda paralelních množin	**Průměrná odpověď**	2,2	2	3,2
	Dolní hranice intervalu spolehlivosti	1	2	1,84
	Horní hranice intervalu spolehlivosti	3,56	2	4,56
Metoda Set Rivers	**Průměrná odpověď**	1,6	2	2,6
	Dolní hranice intervalu spolehlivosti	1	1,12	1
	Horní hranice intervalu spolehlivosti	2,28	2,87	4,27

Z výsledků lze vysledovat, že odpovědi na všechny otázky mají pro metodu Set Rivers menší nebo stejnou průměrnou hodnotu jako pro metodu paralelních množin. Nižší číslo odpovědi zde znamená kladnější hodnocení daného kritéria, takže lze říci, že testovaný vzorek uživatelů subjektivně hodnotil metodu Set Rivers lépe než metodu paralelních množin. Intervaly spolehlivosti se ale pro tyto hodnoty opět velmi překrývají, takže nelze závěr zobecnit.

Průměrné odpovědi s intervaly spolehlivosti jsou znázorněny v následujících grafech (Obrázek 46) stejným způsobem, jako časy plnění úlohy.



(a) *(b)* *(c)*

Obrázek 46: Subjektivní hodnocení metody paralelních množin a Set Rivers účastníky testu

6.7 Zhodnocení výsledků testu

Pomocí uživatelského testu bylo ověřeno, že uživatelé jsou schopni pochopit základní principy vizualizační metody paralelních množin i Set Rivers. Po krátké tréninkové fázi všichni participanti bez problému zvládali základní práci s navrženou aplikací – přesouvání os a kategorií, určení aktivní osy, skrývání a zobrazování os. Také nikomu z participantů nedělalo problém vyčíst z vizualizace základní údaje o datech jako poměry kategorií v rámci osy, poměry dat z kategorií aktivní osy v ostatních kategoriích a popis závislostí dvou os vedle sebe.

U metody paralelních množin se projevily předpokládané problémy s popisem závislostí mezi vzdálenějšími osami. Bez problému dokázali participanti popsat jen tři až čtyři osy vedle sebe. Také bylo u této metody zjištěno, že realizace skládání dimenzí pomocí tvorby vlastní osy není v praxi příliš dobře použitelná, protože je tento způsob pro uživatele zdlouhavý a nekomfortní.

U metody Set Rivers se popsané problémy neprojevily, ale někteří participanti měli problém s tvorbou více filtrů tak, aby rozdělili všechny potřebné svazky. I tak se ale většině participantů pracujících s metodou Set Rivers podařilo dosáhnout lepších výsledků než participantům pracujícím s metodou paralelních množin. S tvorbou více filtrů by uživatelům mohl pomoci delší trénink této metody.

Ze statistik zpracovaných z naměřených časů a dat z dotazníku nelze stanovit žádný závěr, protože testovaných participantů nebylo pro tyto účely dostatečné množství. Uživatelský test měl v tomto případě účel prvního prověření nově navržené metody a nalezení nedostatků. Pro statistické ověření výsledků by bylo vhodné test zopakovat s větším počtem participantů.

7 Závěr

Tato práce se zabývala problematikou vizualizace n-rozměrných heterogenních dat. Byly analyzovány vizualizační metody, které by mohly být pro tento typ dat vhodné a byly popsány jejich nedostatky. Na základě analýzy bylo konstatováno, že n-rozměrná heterogenní data představují pro současné vizualizační metody problém. Poté byla popsána nově navržená technika Set Rivers, která se snaží tento problém řešit.

Dále se v rámci práce podařilo navrhnout a implementovat aplikaci, která umožňuje vizualizaci n-rozměrných heterogenních dat pomocí metody paralelních množin i nově navržené metody. Jedná se o první implementaci metody Set Rivers, takže díky navržené aplikaci lze tuto metodu prakticky ověřit a otestovat. Aplikace umožňuje techniky interakce běžné pro metodu paralelních množin, jako přesouvání os a kategorií, skrývání os, editaci kategorií i skládání dimenzí prostřednictvím uživatelské osy. Dále umožňuje techniky navržené v rámci metody Set Rivers jako zobrazení rovnoběžníků uvnitř kategorií os, zobrazení číselné osy a filtrování dat.

V rámci práce byl proveden a reportován také uživatelský test, pomocí kterého byla ověřena funkčnost aplikace i nově navržené metody a dvě použité vizualizační metody byly srovnány. Na základě výsledků testu lze říci, že uživatelé jsou schopni bez problému s aplikací pracovat a pochopit základní principy obou vizualizačních metod. U metody paralelních množin se projevily předpokládané nedostatky, které se u metody Set Rivers neprojevily. Určité problémy byly identifikovány i u metody Set Rivers, ale i tak se většině participantů testu podařilo s touto metodou dosáhnout lepších výsledků.

Provedený uživatelský test splnil účel prvního prověření nové metody a popsání nedostatků. Jako možné pokračování této práce by bylo vhodné provést testování s větším počtem participantů, ze kterého by bylo možné statisticky zpracovat výsledky.

8 Literatura

[1] ANDREWS, David F. Plots of high-dimensional data. Biometrics, 1972, 125-136.

[2] BECKER, Richard A.; CLEVELAND, William S. Brushing scatterplots. *Technometrics*, 1987, 29.2: 127-142.

[3] BENDIX, F., R. KOSARA a H. HAUSER. Parallel sets: visual analysis of categorical data. In: *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005* [online]. IEEE, 2005, s. 133-140 [cit. 2017-01-20]. DOI: 10.1109/INFVIS.2005.1532139. ISBN 0-7803-9464-X. Dostupné z: http://ieeexplore.ieee.org/document/1532139/

[4] DIACONIS, Persi; FRIEDMAN, Jerome H. M and N plots. Department of Statistics, Stanford Univ., 1980.

[5] D'OCAGNE, Maurice. *Coordonnées parallèles & axiales: méthode de transformation géométrique et procédé nouveau de calcul graphique déduits de la considération des coordonnées parallèles*. Gauthier-Villars, 1885.

[6] HARTIGAN, John A. Printer graphics for clustering. Journal of Statistical Computation and Simulation, 1975, 4.3: 187-213

[7] HEINRICH, J. a D. WEISKOPF. *State of the Art of Parallel Coordinates* [online]. 2013 [cit. 2017-01-20]. Dostupné z: http://dx.doi.org/10.2312/conf/EG2013/stars/095-116

[8] HOFMANN, Heike. Exploring categorical data: interactive mosaic plots. *Metrika*, 2000, 51.1: 11-26.

[9] HOWELL, David C. Statistical methods for psychology. 7th ed. Belmont, CA: Thomson Wadsworth, c2010.

[10] CHERNOFF, Herman. The Use of Faces to Represent Points in K-Dimensional Space Graphically. *Journal of the American Statistical Association* [online]. 1973, **68**(342), 361- [cit. 2017-12-04]. DOI: 10.2307/2284077. ISSN 01621459. Dostupné z: http://www.jstor.org/stable/2284077?origin=crossref

[11] JOBLOVE, George H.; GREENBERG, Donald. Color spaces for computer graphics. In: *ACM siggraph computer graphics*. ACM, 1978. p. 20-25.

[12] KOSARA, Robert; BENDIX, Fabian; HAUSER, Helwig. Parallel sets: Interactive exploration and visual analysis of categorical data. *IEEE transactions on visualization and computer graphics*, 2006, 12.4: 558-568.

[13] KOSARA, Robert; SAHLING, Gerald N.; HAUSER, Helwig. Linking scientific and information visualization with interactive 3D scatterplots. 2004.

[14] LIKERT, Rensis. A technique for the measurement of attitudes. *Archives of psychology*, 1932.

[15] MORRIS, Christopher J.; EBERT, David S.; RHEINGANS, Penny. An Experimental Analysis of the Effectiveness of. *Features in Chernoff Faces. AISPRS99, University of Maryland Baltimore County*, 1999.

[16] Mplot3d. *Matplotlib* [online]. [cit. 2018-01-07]. Dostupné z: https://matplotlib.org/mpl_examples/mplot3d/scatter3d_demo.png

[17] NEYMAN, Jerzy. Outline of a theory of statistical estimation based on the classical theory of probability. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 1937, 236.767: 333-380.

[18] MUNZNER, Tamara. *Visualization analysis and design*. CRC press, 2014.

[19] NÚÑEZ, Reyes; JESÚS, José. Ideas for the use of Chernoff faces in school cartography. 2009.

[20] Scatterplots. *Quick-R* [online]. [cit. 2018-01-07]. Dostupné z: https://www.statmethods.net/graphs/images/spmatrix1.jpg

[21] Scatterplot Matrix Brushing. *Mike Bostock's Blocks* [online]. [cit. 2018-01-07]. Dostupné z: https://bl.ocks.org/mbostock/4063663

[22] SHAW, Christopher D., et al. Using shape to visualize multivariate data. In: *Proceedings of the 1999 workshop on new paradigms in information visualization and manipulation in conjunction with the eighth ACM internation conference on Information and knowledge management*. ACM, 1999. p. 17-20.

[23] TURNER, Eugene. Life in Los Angeles 1970. *California State University Northridge* [online]. [cit. 2018-01-07]. Dostupné z: www.csun.edu/~hfgeg005/eturner/images/Maps/lifeinla.gif.

[24] Usability tests. *Přednáška předmětu Testování uživatelského rozhraní* [online]. [cit. 2017-12-20]. Dostupné z: https://cent.felk.cvut.cz/courses/Y39TUR/?page=slides

[25] UTTS, Jessica M. *Seeing through statistics*. 3rd ed. Belmont, CA: Thomson, Brooks/Cole, c2005, s 166-167. ISBN 05-343-9402-7.

[26] Visualization of multi-dimensional data. *Přednáška předmětu Vizualizace* [online]. [cit. 2017-12-07]. Dostupné z: https://moodle.fel.cvut.cz/pluginfile.php/59230/mod_label/intro/07_Visualization_of_n-dimensional_data.pdf

[27] WARD, Matthew O. Multivariate data glyphs: Principles and practice. In: *Handbook of data visualization*. Springer Berlin Heidelberg, 2008. p. 179-198.

[28] *XDAT – A free parallel coordinates software* [online]. [cit. 2018-01-08]. Dostupné z: https://www.xdat.org/

[29] ZARANKIEWICZ, Casimir. On a problem of P. Turán concerning graphs. *Fundamenta Mathematicae*, 1955, 41.1: 137-145.

Příloha A – úkoly pro uživatelský test

Seznámení s vizualizačním nástrojem

Jste analytik/analytička cestovní kanceláře a vaším úkolem je analyzovat data o pokojích ve všech nabízených hotelech v Římě.

Ukázka

Najděte všechny pokoje se snídaní v ceně v hotelech bez hvězdičky (stars = undefined). Můžete popsat závislosti mezi jednotlivými atributy?

Úkol

Najděte všechny 2 lůžkové pokoje v hotelech bez hvězdičky (stars = undefined). Můžete popsat závislosti mezi jednotlivými atributy?

Test

Vaším úkolem je nyní analyzovat pokoje v hotelech v Brně.

Úkol

Najděte všechny 4 lůžkové pokoje. Jaké jsou relace/závislosti mezi atributy pro tyto pokoje?

Příloha B – dotazník pro uživatelský test

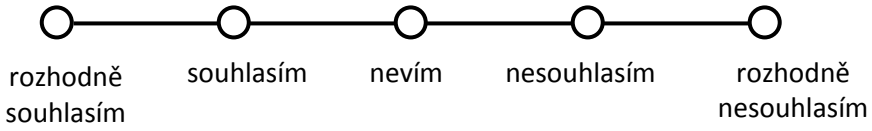
Post-study dotazník

Participant: Věk:

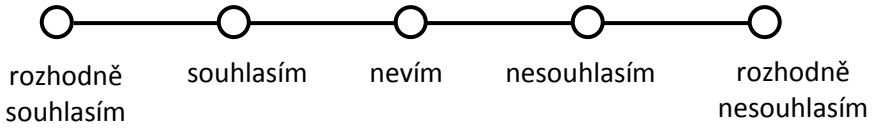
Datum:

Metoda: M1 M2

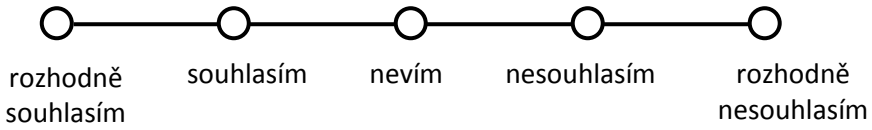
Metoda pro mě byla komfortní.



Metoda mi dovolila pracovat rychle.



V plnění úkolů jsem si byl jistý.



Neformální zpětná vazba: