

GEOGRAFICKÁ DATA

- Data, která jsou mix abstraktních a prostorových dat
 - Obsahují nějakou geometrii – body, linie, oblasti (linie – řeky, oblasti – země, města)
 - Obsahují abstraktní data, která souvisí s geometrií (jméno řeky, populace země)
- Data, která mají nějakou geografickou lokaci (zeměpisnou šířku/výšku, poštovní číslo, adresa, ..)
 - Mají kvantitativní atributy = teplota, směr větru, rychlost (obohacení dat možné)
 - Mají nominální atributy = typ kriminality, politická skupina pro kterou volili, ...
 - Je potřeba viz. založené na frekvenci
- Např.
 - Mapy počasí – máme kvantitativní atributy jako teplota a srážky, která měříme na nějaké geografické lokaci
 - Události (smrt v relaci na cholera) s adresami, geografické lokace vodních nádrží

Z hlediska primitiv, na které můžeme mapovat geografická data:

POINTS – BODY

- Pokud máme kvantitativní atributy změřené na geografické lokaci, tak se můžeme chovat jako kdyby to byly scattered polohová data
 - Udělej mřížku s buňkami vytvořenými triangulací
 - Rekonstruuj spojitý prostor pomocí datového obohacování
 - Namapuj na vizuální kanál (např. barvu)
- Pokud máme nominální atributy získané z geografické lokace, tak neobohacujeme
 - Například:
 - Spojené s typem místa našeho zájmu (letišť, monument, atd.)
 - Seznam lidí, co umřeli na cholera se seznamem jejich adres (tady třeba počítáme úmrtí na dům a nedává smysl interpolovat mezi domy)
 - Seznam nezaměstnaných s jejich adresami
 - Dělá se vizualizace založená na frekvenci
 - Spočítáme frekvenci dat v polohovém prostoru a vizualizujeme tyto frekvence

LINES A AREAS – ČÁRY A OBLASTI

- Lomenné čáry (polylines)
 - Sekvence bodů s geografickou lokací
 - Typicky reprezentujeme takto silnice, řeky, trajektorie, atd.
 - K liniím se můžou vázat neprostorové atributy, které se vztahují na:
 - Celou linii (typ silnice – dálnice, lokální silnice, atd.)
 - Segment linie (hustota dopravy na silnici v segmentu)
 - K jednotlivým bodům na lomenné čáře – pro kvantitativní atributy můžeme interpolovat na této lomenné čáře
- Oblasti
 - Uzavřené sekvence bodů s geografickou lokací
 - Reprezentují čtvrtě, národní parky, lesy, země, kontinenty, ...
 - Typicky máme neprostorové atributy vztahující se k celé oblasti
 - Často jsou atributy jen výsledkem agregace bodových dat v oblasti

Vizualizace na základě frekvence

- Máme neprostorová data a neprostorové atributy
- Zajímají nás nejvíce neprostorová data (např. nezaměstnanost), ale chceme tato neprostorová data nějak zpracovávat v rámci těch prostorových (na regiony)
- Například: máme dataset nezaměstnaných – máme jejich adresu, jméno a míra nezaměstnanosti
 - o Nejvíce nás zajímá míra nezaměstnanosti, ale chceme vzít v potaz taky lokaci
 - o Data jsou agregována vzhledem na adresy pro geografickou lokaci (pro každý region spočítám míru nezaměstnanosti)

Typické tasky pro geografickou vizualizaci

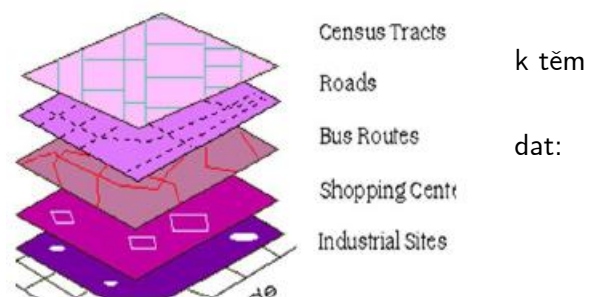
- *Identifikuj* – scope je single target
 - o Chceme získat charakteristiky targetu z vizualizace
- *Compare* – scope je několik targetů

Tasky spojené s geometrií

- *Tasky se vzdáleností* – domy, které jsou blízko mého pracoviště a školy?
 - o Dynamické queries
- *Tasky s trasou*
 - o Jaká je nejkratší cesta z A do B?
- *Tasky s pohybem či flow*
 - o Kolik lidí migruje z LA do NY
 - o Mezi jakými městy lidé migrují nejvíce

REPREZENTACE GEOGRAFICKÝCH DAT

- prostorová data (geometrie)
- neprostorová data (atributy polohových dat)
- Polohová data: body, čáry, oblasti
- Nepolohová data: podobná tabulárním datům, vztahují se polohovým
- Data jsou v několika vrstvách, které reprezentují nějaký typ
 - o Kvantitativní data - výška plochy
 - o Nominální – typ bodu, čáry, area features
 - o Vrstva může mít jen jednu (pouze řeky), nebo několik atributů (okresy, typy okresů)



Mapové vrstvy mají podobné vlastnosti.

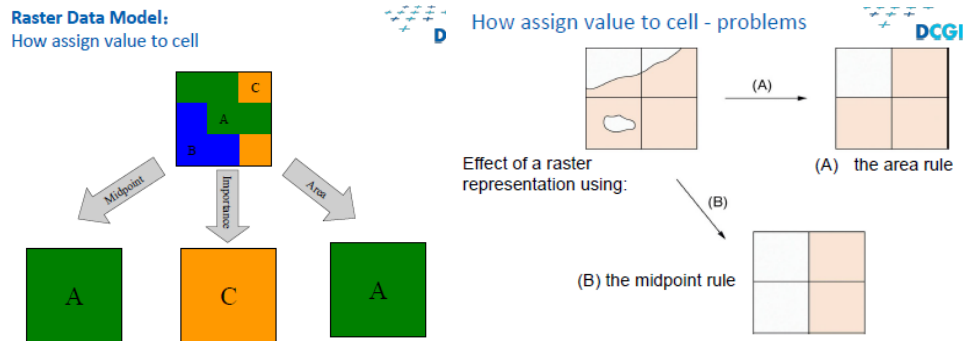
Jsou dva způsoby jak reprezentovat geospaciální data – rastrový a vektorový model.

Často se kombinují.

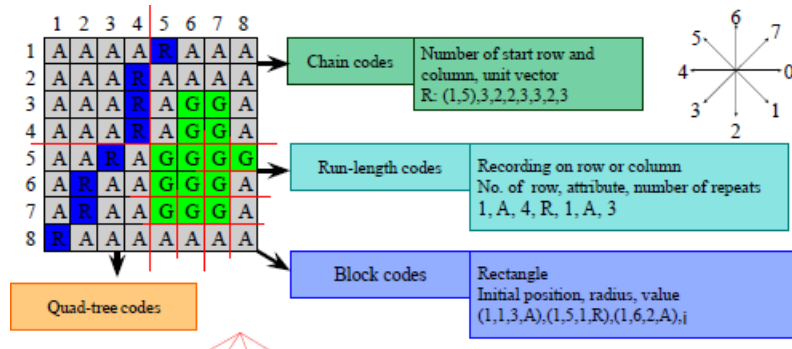
RASTROVÝ MODEL

- Využívá 2D rastrovou grafiku – rozdělí celou vrstvu na regularní mřížku, každá buňka obsahuje jednu hodnotu
- V rastrové datové struktuře:
 - o Bod: pixel nebo buňka
 - o Linie: reprezentováno jako cluster sousedných pixelů
 - o Oblast: pixely s těmi samými atributy
- Když vytváříme rastrový model, tak máme několik možností, jak přiřadit hodnotu do jednotlivých buněk nebo pixelů:
 1. Vezmeme vzorek ve středu buňky a do té buňky dáme hodnotu, co se tam nachází (midpoint)
 2. Oblasti mají důležitost – vložíme nejdůležitější (importance rule)
 3. Jakou část buňky zabírá – vložíme to, co zabírá nejvíce místa (area rule)

Tento přístup může způsobit problémy.

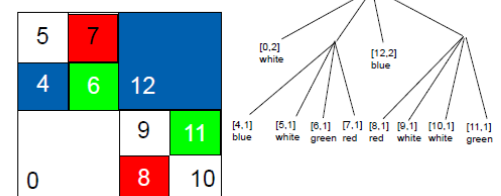


- Jsou jiné způsoby, jak úsporněji reprezentovat:
 - o Řetězové kódy, vhodné zejména pro reprezentaci linií
 - Na začátku máme „souřadnice“, kde máme začít – řádek a sloupec
 - Pak postupujeme po číslech, který reprezentují směry (ta hvězdička, modrý R)



- o Run-length kódy – máme na jakém řádku jsme, pak jaký symbol se tam nachází a kolikrát se bude opakovat (př. řádek 1, dáme do buňky A a opakuje se 4x, pak máme R, opakuje se 1x, ...)
- o Block kódy – snažíme se data reprezentovat jako sérii čtverců (jsme na souřadnicích 1,1, výška a šířka je 3x3, vyplníme písmenem A)
- o Kvadrantové stromy – rozdělíme vrstvu na stejně velké oblasti a pokud kvadrant není uniformní, tak ho zase dělíme, dokud není kvadrant uniformní
 - Záleží na uniformitě, jak ten strom bude vysoký
 - Dva typy kvadrantových stromů:
 - Region quadtree
 - o Uchovává informaci o stromech jen v listech stromu

Quad-Tree Representation

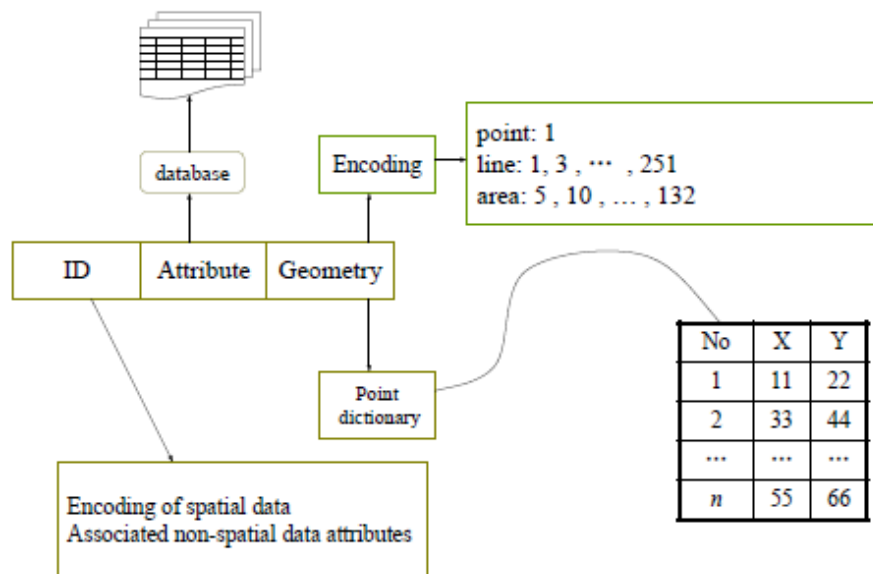


- Pyramida

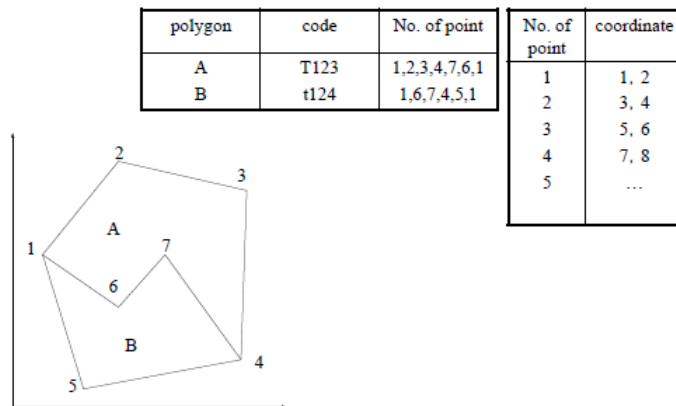
- Dodatečná informace ve vnitřních uzlech – např. pokud máme v uzlech uloženou jednu informaci o tom, zda se u měst nachází řeky, tak lze jen uložit do vnitřních uzlů, kolik potomků on má (a tím pádem víme kolik měst má řeku, atd.)
- Hierarchii stromu pak můžeme zobrazovat na různých úrovních, protože máme dodatečnou informaci o uzlech

VEKTOROVÝ MODEL

- Body, čáry, oblasti (polygony), topologický relace
- Pouze pro zaznamenání pozice a informace o atributech polohových objektů
- Existují reprezentace s topologickými vztahy i bez
- využívá nějaké databáze (storage), kam se odkazuje, když potřebuje souřadnicový systém (je separovaný od prostorových objektů)
- Vlastnosti:
 - Bez topologických relací (zatím) – jen jednoduché dotazy na prostorové objekty
 - Datová redundance – linie, co sharují dvě oblasti jsou reprezentované 2x
 - Prostorová analýza je více komplexní (protože nemáme tu topologickou informaci)



- Point je reprezentován jen 1 bodem, linie nějakou sekvencí bodů a oblasti také s tím, že se očekává, že na konci bychom se zase vrátili zpátky (uzavřená oblast)
- Redundanci vidíme u linie 1, 6, 7, 4 – je jak v polygonu A, tak v polygonu B

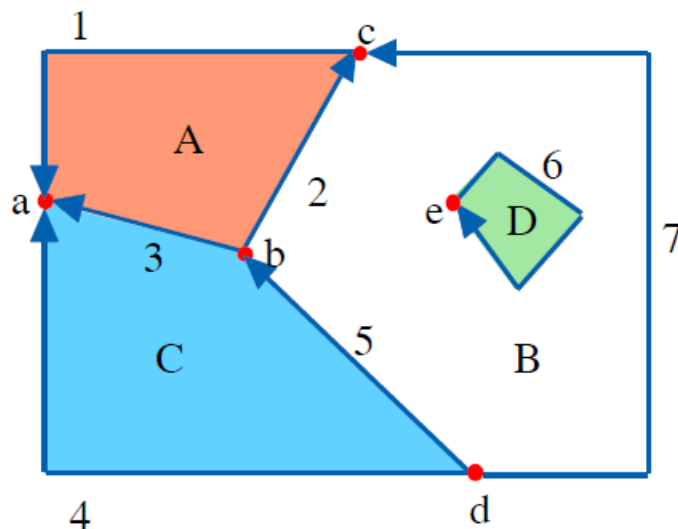


Topologické relace ve vektorovém modelu

- Topologie = jak matematicky definovat relace, nás zejména zajímá relace sousednosti
- Budeme k tomu používat arc-node model
 - o Node = body, ve kterých se dva nebo více oblouků potkávají
 - o Arc = série bodů, který začínají a končí na uzlu

Arc ID	Left Poly	Rt Poly	From node	To node
1	A		c	a
2	A	B	b	c
3	C	A	b	a
4		C	d	a
5	C	B	d	b
6	B	D	e	e
7	B		d	c

Poly ID	No. of arcs	List of arcs
A	3	-1, -2, 3
B	4	2, -7, 5, -6
C	3	-3, -5, 4
D	1	6



Vlastnosti, které vznikají díky topologii:

Connectivity: arcs jsou spojené s jinými pomocí nodes -> identifikuje možné trasy, networky (řeky, silnice – vyhledávání) přes list arcs a nodů v databázi

Containment: Uzavřený polygon má oblast, která se dá měřit obsah (to jsme mohli i před topologií); arcs definují hranice a uzavřené oblasti

Contiguity: sousednost polygonů může být určena pomocí arců – to můžeme vyšetřovat, které jsou sharované

Toto je fundamentální pro GIS analýzu a queries, např.:

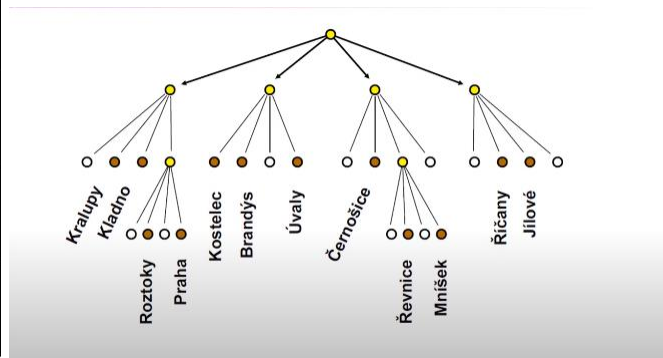
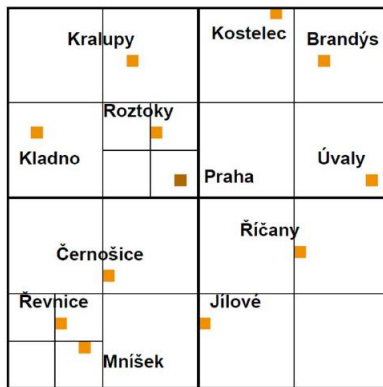
Jak se dostanu z A do B pomocí nějakého systému silnic?

Jaká je plocha kombinované obyvatelné oblasti?

Jaká obytná oblast je vedle parků?

Akcelerace datové struktury

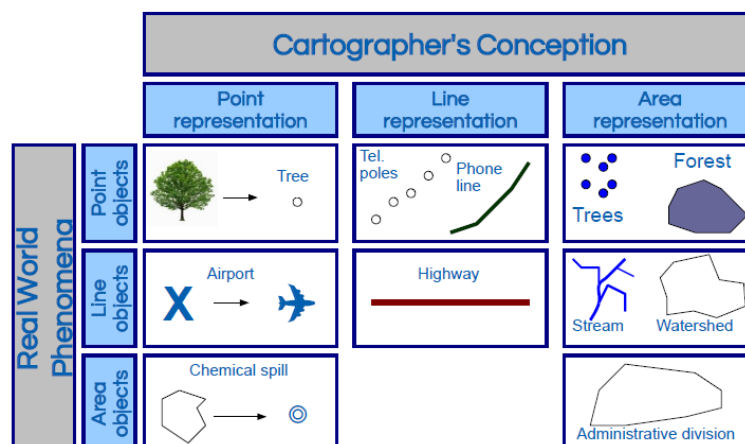
- Pomáhá nám k najít bodů, aniž bychom museli procházet všechny body, které máme k dispozici
- PR quadtree



- Zase splitneme kvadranty na uniformní kvadranty tak, aby v každém byl jeden bod
- Zrychlení výpočtu, které body se nachází v regionu našeho zájmu:
 - o Definujeme si nějaký obdelník, který reprezentuje místo, co nás zajímá a díky naší akcelerrační struktuře už nemusíme testovat každý bod, jestli se nachází uvnitř obdelníku nebo ne
 - o Jednotlivé kvadranty testujeme, zda se protíná s obdelníkem – pokud ano, tak testujeme menší a menší kvadranty
- Zase informace jen v listech

Vizualizace map

- Mapování reálných objektů na geometrii záleží na typu, důležitosti, úrovni detailu, atd.



Vizuální kanály, na které můžeme mapovat

- Velké množství zdrojů
- Záleží na informaci, kterou chceme mapovat

POINT FEATURES = barva, tvar, velikost

LINE FEATURES = barva, styl, šířka

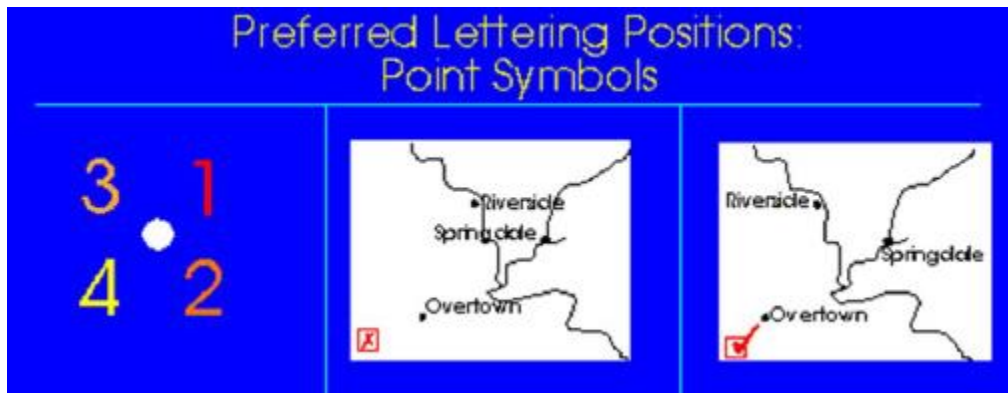
AREA FEATURES = barva, textura

Typografie

- Správné umístění a vzhled text informací
- Velice těžký task (NP-úplná úloha)
- Kartograf musí řešit obsah i formu
 - o Obsah (content) je relevance a jasnost textu
 - Musí se vyvarovat zmatku a špatného vyložení
 - Redundance
 - Zkratek
 - o Forma je vzhled textu na mapě

Labelování bodů

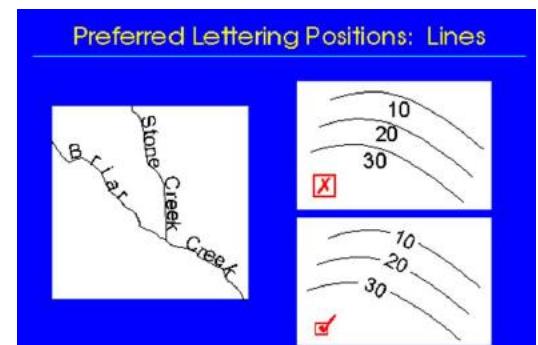
- Písmena na hranicích jsou problémy
- Jsou pouze 4 místa relativní vůči textu, kam lze text umístit
 - o Některé jsou lepší než druhé (preferuj 1, pak 2..)
- Body nemají orientaci, takže text musí být horizontálně



- Čáry a oblasti
 - o Text by měl sledovat orientaci (směr) čáry (měl by se deformovat)
 - o Text by měl vyplnit oblast (měl by být uvnitř)
 - o Měl by protínat vrstevnici

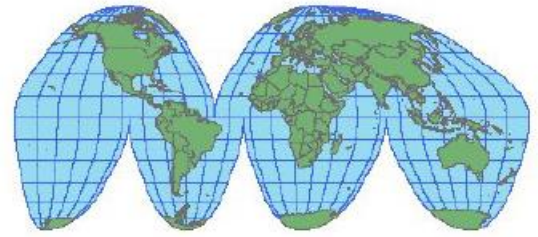
Projekce map

- V mapách nás zajímají 3 vlastnosti – tvar, úhel a velikost
- Země je kulatá, 2D projekce ji vždy deformuje nějakým způsobem – je potřeba hodně projekcí
 - o Odlišný projekce nám ukáží rozdílné vlastnosti, jsou více či méně přesne
 - o Nejčastěji je deformovaná velikost



Mercator projekce – nejpoužívanější

- Deformuje značně – velikosti oblastí
- Zachovává úhly mezi rovnoběžkami a poledníky
- Čím dál od rovníku, tím větší se daná oblast jeví
 - o *Afrika je na rovníku, takže její velikost není scuffed, ale USA, Evropa jsou dál, takže je jejich velikost scuffed*



Goode Homolosine Projection

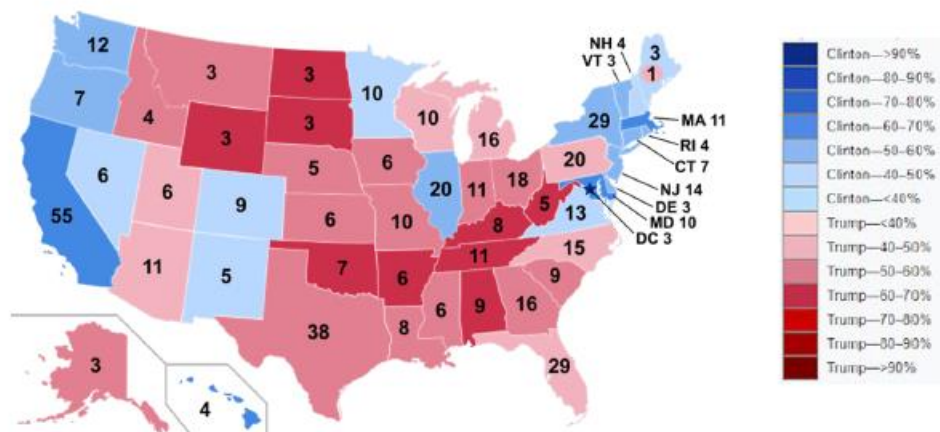
Goode Homolosine Projekce

- Snaží se zachovávat velikost
- Ale zase deformuje úhly mezi rovnoběžky a poledníky

Vizualizace geografických dat

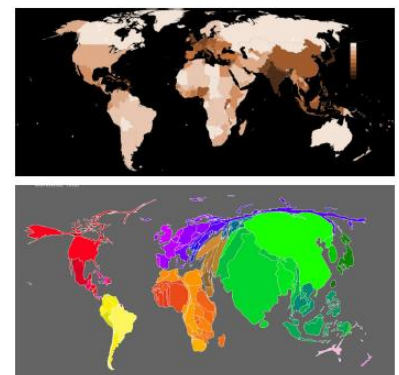
Choropleth maps

- Mapování spojitého kvantitativního atributu na odstín a jas
- Odstín můžeme určit na základě druhého nominálního atributu
 - o Clintonová vs Trump: odstín je dán tím, kdo vyhrával a jas je na tom, o kolik



Kartogramy

- Deformuje velikost oblastí na základě hodnot atributů dat
- Velikost může být již deformována kvůli projekci předchozí (takže bychom měli vědět, jaká projekce byla předtím použita)
- Deformace není vnímaná úplně přesně
 - o Je velmi těžké odhadnout, jak moc je to deformované
 - o Porovnávání hodnot je těžké
- Schopné ale lépe vizualizovat rozdíly mezi hodnoty než Choropleth (rozdíly jsou zvětšené)



Choropleth and cartogram of population density

Glyph

- Můžeme používat zase glyfy, které se používají na tabulková data
 - o Star glyf, stick figures, Chernoff faces
- Výhoda je, že můžeme namapovat několik atributů dat a pak je umístit na geografickou mapu

Vizualizace trajektorií

- Pohyb z bodu A do B
- Máme dvě možnosti:
 - o Nezajímá nás exaktní tvar trajektorie
 - Let z jednoho města do druhého, migrace lidí
 - o Zajímá nás tvar trajektorie
 - Pohyb po zakřivené čáře – cesta autem z jedné adresy do druhé
- Trajektorie (hrany nebo arcs) formují network mezi body (nody), takže lze chápat jako relační data
- Můžeme aplikovat vizualizaci pro networks s tím, že máme požadavek navíc:
 - o Nody mají geografickou pozici
 - o Vizualní kanál pro 2D pozice už je použita na lokaci, takže nemůžeme mapovat žádná data na pozici

Frequency-based vizualizace hran

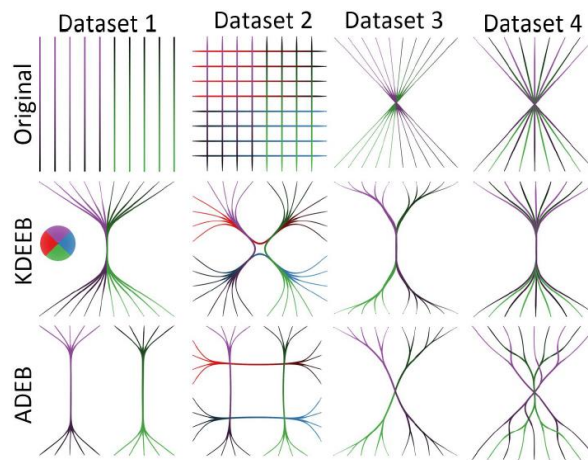
- trajektorie reprezentují pohyb z A do B
 - o nezajímá nás exaktní tvar trajektorie
- zajímá nás proporce
- potřebujeme snížit visual cluster
 - o způsobený překřikováním hran
 - o nemůžeme předělávat pozice, protože nody mají svojí geografickou lokaci
- technika Edge Bundling
 - o pro redukci visual clutter
 - o ty hrany/oblouky, které mají podobný směr a pozici svážeme dohromady – podobná myšlenka jako svazování kabelů
 - o existuje několik technik:
 - hierarchické edge bundling
 - grid-based (implementačně náročná), force-directed, attribute-based, ...

Force-directed edge bundling

- hrany jsou rozdělené do segmentů
- založená na iterativnímu přístupu
 - o definujeme sílu mezi korespondujícími body na hranách
 - o počítáme, jak síla deformuje ty hrany
 - o síla působí tak, že sousední body se přitahují a přitažlivá síla je proporcionální vzdálenosti
 - čím dál, tím víc je to přitažlivé
 - pro hrany, které jsou blízko sebe – síla bude působit víc na ně, než na hrany, jejichž body budou daleko od sebe
- kompatibilita hran je evaluována – hrany, který jsou kompatibilní jsou svázány dohromady
- ne vždy chceme totiž hrany svázat dohromady – hlavně, když mezi sebou mají velký úhel, což může být ošetřeno kompatibilitou
- děláme dokud se ten pohyb neustálí

Attribute based edge bundling

- ještě více je evaluována kompatibilita hran



- např. máme orientaci hran, to chceme oddělit, nebo jiná vlastnost jako čas atd.
- Stejně ale nemusí být moc dobře vidět, kam ty hrany vedou

Untangling origin-destination flows

- Tohle se snaží řešit agregací hran do svazků a pak barevným mapováním, abychom věděli co kam jde

