Algoritmy vektorizace leteckých snímků

Jaroslav Fojtík

Czech Technical University, Faculty of Electrical Engineering
Center for Machine Perception
121 35 Praha 2, Karlovo náměstí 13, Česká Republika
telefon +42 2 24357465, FAX +42 2 290159
Email:fojtik@vision.felk.cvut.cz

1. Úvod

Jednou ze snah kartografů je snaha o udržení aktuálnosti jednotlivých map. Za tímto účelem jsou prováděna nová letecká snímkování terénu a podle výsledných fotografií jsou upravovány staré mapy. Dosud je tento typ práce většinou dělán ručně. Snahou autora je prozkoumání možnosti algoritmizace jednotlivých kroků. Výsledkem má být úspora lidské práce při tvorbě map z leteckých snímků.

Tato práce se zabývá převodem rastrových dat do vektorové podoby. Dosavadní postupy vektorizace s nimiž autor měl možnost se seznámit nedávaly pro případ leteckých snímků uspokojivé výsledky. Proto byl navržen a realizován nový postup založený na topologii konečných prostorů.

Zpracování leteckého snímku probíhá v následujících krocích:

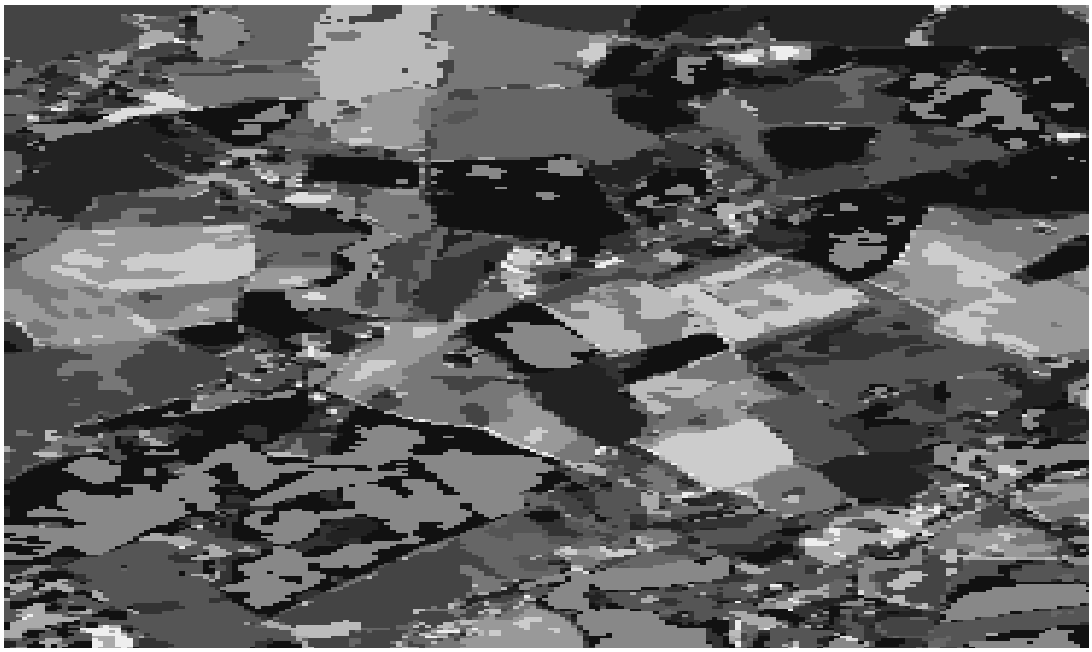
- Klasifikace (není součástí této práce)
- Obraz s obarvenými oblastmi
- Hranové body
- Uzlové body
- Vektorový popis hran
- Tvorba ploch a kontrola jejich obsahu

2. Vstupní data

Při leteckém snímkování vzniká bitmapový letecký snímek. Tento je potřeba předzpracovat libovolným klasifikátorem např neuronovou sítí. Návrh klasifikátoru a výběr příznaků není součástí této práce. Hlavní požadavek kladený na vstupní data vektorizátoru je existence shluků pixelů se stejným typem obarvení.

Algoritmus vektorizace je navržen pro vyhledávání hranic souvislých oblastí. Existují v něm homogenní oblasti se stejným označením (reprezentovaným barvou). Každé označení představuje rozhodující vlastnost vstupních dat. Např les, jezero, silnice atd. Vyhledané hranice oblastí jsou následně převáděny do vektorové reprezentace.

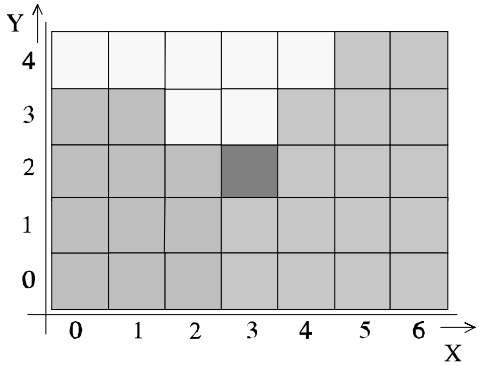
Popisovaný algoritmus není navržen pro hledání hran v šedotónovém obraze. Pro tuto činnost je však možné data předem předzpracovat. Hrany mohou být nalezeny nějakým hranovým detektorem, např Canny, a jeho výstup (po prahování s hysterezí) pak již lze dále převádět níže popsaným postupem do vektorové reprezentace.



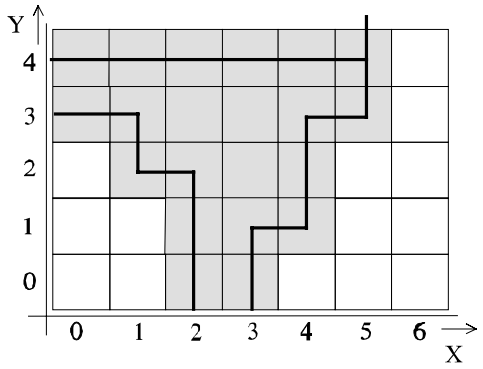
Obr. 1 Příklad vstupního obrázku

3. Dřívější metody vektorizace oblastí

Celý obraz je procházen operátorem, který testuje rovnost hodnot dvou sousedních pixelů. Rovnost je testována ve všech směrech 8 okolí aktuálního bodu (popř 4 okolí). Pokud dojde v některém směru nebo ve více směrech současně k nerovnosti, pak je ve výsledném binárním obrázku pixel nastaven na hodnotu 1. Jinak je roven 0. Postup je uveden na následujících obrázcích:



Obr. 2 Původní bitová mapa



Obr. 3 Bitmapa po vektorizaci

Tento postup však má podstatnou nevýhodu dvojité odezvy na každou hranu. Existují samozřejmě vylepšené varianty, které testují shodu pouze v několika směrech (pouze zprava doleva a shora dolů). Tím je sice částečně kompenzována dvojitá odezva na změnu. Ne však úplně. Problémy se projeví při výskytu osamoceného bodu na nějaké hraně. Navíc dojde k nežádoucímu posunutí všech hran v určitém směru.

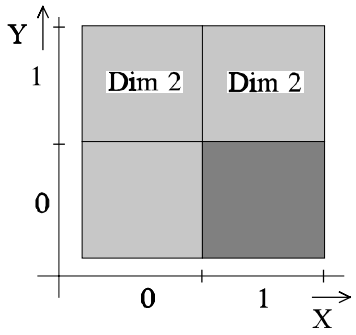
Předcházející problémy vycházejí z nesprávné reprezentace hranic oblastí. Proto jsou využity výsledky topologie konečných prostorů [Kovalevski].

© December 12, 1997

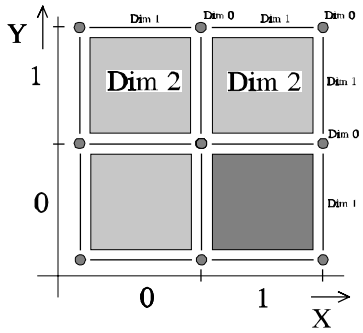
4. Popis metody abstraktních celuárních komplexů

Při pokusu o hledání hranic oblastí a jejich reprezentaci není původní bitová mapa příliš vhodná. Proto byla použita nová metoda, která zajistí mnohem lepší kvalitu detekovaných hran.

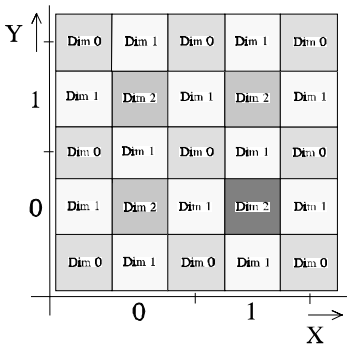
Do obrázku jsou uměle přidány nové body v rozích a sousedních hranách původních bodů. Tyto nové body pak mají speciální vlastnosti a slouží k vyjádření topologických vazeb mezi obrazovými daty. Postup rozšíření je názorně ukázán na následující sérii obrázků.



Obr. 4 Původní obrázek



Obr. 6 Celuární komplexy



Obr. 5 Nová reprezentace

Do bitové mapy jsou vloženy nové body. Ty popisují hrany pixelů a rohové body. Při uvedené konfiguraci již nevzniká paradox dvou přímek [viz Počítačové Vidění str:30].

Původní pixely představují elementy s dim 2. Hranice dvou pixelů mají dim 1 a elementy ležící v rozích pixelů mají dim 0.

Hrana jakožto 1D útvar nemůže samozřejmě přes body s dim>1, může procházet pouze body s dimenzí <=1. K větvení hrany (uzlu) může dojít pouze bodě s dim 0, tzn pouze v rozích dvou pixelů.

5. Algoritmus vektorizace

Vektorizátor má 5 hlavních průchodů. Při prvních dvou jsou zpracovávána data pouze v rastrové podobě, při třetím je vytvářena vektorová reprezentace a poslední dva pracují pouze s vektorovou reprezentací:

6. Popis jednotlivých průchodů

1.Průchod - **hledání hranic**

V tomto průchodu jsou hledány hranice oblastí. Je vytvářen nový pomocný binární rastr KDT.RAS s rozměry [2*x+1;2*y+1]. Jeho okraje jsou nastaveny na 1. Vnitřní body jsou nastaveny pouze za podmínky, že leží mezi dvěma body s rozdílnou úrovní jasu. Je vyhodnocován rozdíl v podélném, příčném směru a na úhlopříčkách.

původní obrázek

1	1	2
1	1	2
4	1	3

je transformován na

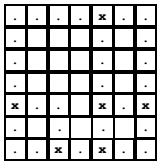
.	.	.	.	.	.	.
.	1		1	.	2	.
.				.		.
.	1		1	.	2	.
.	.	.			.	.
.	4	.	1	.	3	.
.	.	.	.	.	.	.

(v bitové mapě jsou uloženy pouze body označené v tabulce tečkami, čísla jsou uvedena pouze pro zvýraznění korespondence s předchozím obrázkem)

2.Průchod - **hledání uzlů**

V tomto průchodu jsou detekovány uzly. Je vytvořena druhá pomocná bitová mapa uzlů KDU.RAS s rozměry [2*x+1;2*y+1]. Uzel je takový bod, který má více než dva nenulové sousední body. Pro detekci uzlů je využit binární operátor podobný dilataci.

Takže z našeho obrázku vznikne:



(v bitové mapě jsou uloženy pouze body označené v tabulce písmenem 'x', tečky jsou uvedeny pouze pro zvýraznění korespondence s předchozím obrázkem)

3.průchod - **detekce vektorových linií**

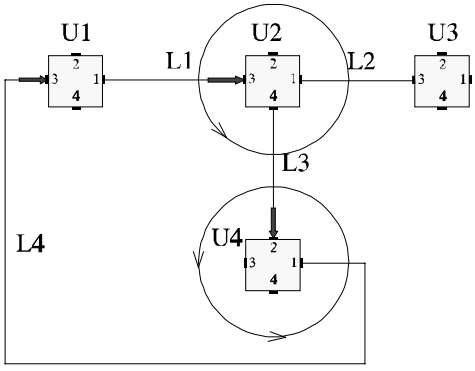
V tomto průchodu jsou počítány vektorové linie. Každá vektorová linie musí vycházet z nějakého uzlu a do nějakého (jiného) uzlu vstupovat. Jsou vytvářeny 3 další pomocné soubory BODY.BIN, UZLY.BIN a LINIE.BIN. Vyhledávání linií probíhá následujícím způsobem: Procházím tak dlouho sekvenčně obrázek až narazím na první uzel. Z něho se pokusím postupovat postupně všemi čtyřmi směry. Při postupu jsou hledáni následníci bodů a ti jsou zaznamenáváni do interní struktury. Při postupu jsou body postupně z rastru vymazávány. Takto se pokračuje až do nalezení uzlového bodu. Potom je linie tvořena seznamem lomových bodů. Počet lomových bodů je optimalizován speciálním algoritmem.

Z předchozího je patrné, že uvedený postup neumožnuje nalezení uzavřených kružnice bez lomových bodů. Proto je 3 průchod proveden 2x. Při druhém postupu je kterýkoliv nalezený bod považován za uzlový (s řádem 2) a je postupným procházením nalezena celá kružnice. (Vše ostatní již bylo z rastru KDT.RAS vymazáno!)

Na konci průchodu jsou dále již nepotřebné rastry KDU.RAS a KDT.RAS smazány.

4.průchod - **kontrola ploch**

Z každého uzlového bodu je postupováno na linii, následuje nový uzlový bod. Poté vždy proti směru hodinových ručiček až bude dosaženo do počátečního bodu. Ze zadání (vzhledem k předchozím krokům) vyplývá, že graf musí být rovinný. Z toho vyplývá, že výše popsaným způsobem je vždy uzavřena kružnice.



Obr. 7 Procházení ploch

Příklad postupu hledání:

Nechť je výchozí bod U1 a tento uzel má obsazenu přípojku 1. Tedy je postupováno po linii L1 do uzlu U2. Do uzlu je vstoupeno přípojkou č.3. První obsazená přípojka ve směru hodinových ručiček je č.4. Po této přípojce je uzel nakonec opuštěn. Algoritmus postupuje po linii L3 do uzlu U4. Do uzlu U4 je vstoupeno přípojkou č.2. První obsazená přípojka ve směru hodinových ručiček je č.1. Výstupní linie L4 se již vrací do vstupního uzlu U1. V

© December 12, 1997

tomto kroku tedy plocha uzavřena a algoritmus končí.

Při tom je ovšem současně počítána velikost plochy. Je-li větší, než stanovený limit, pak jsou všechny linie, které k ní patří označeny za platné (poznámkou v uzlových bodech).

V uzlových bodech jsou poznamenávány všechny informace o postupu. Tím je zajištěno, že plocha nemůže být nalezena 2x.

5.průchod - **vytváření topologické struktury**

Je postupováno, podobně jako v předchozím průchodu po jednotlivých uzlech. V každém uzlu jsou příslušné nevyřazené linie topologicky spojeny dohromady.

V tomto průchodu je prováděno i slepování zbylých linií. Některé linie mohou být slepeny pouze proto, že byli jejich sousedi vyřazeni při předchozím průchodu a vznikl uzel řádu 2. Každý uzel řádu 2 (s výjimkou kružnic) je vyhledán a vyřazen z grafu. Při vytváření linií jsou též generovány plochy.

Na konci průchodu jsou dále již nepotřebné soubory LINIE.BIN, BODY.BIN a UZLY.BIN smazány.

7. Návod pro uživatele

Program má celkem 5 parametrů:

a, Jméno vstupního obrázku.
b, Jméno výstupního bloku.
c, Minimální plochu.
d, Odchylku vektorové linie od rastru.
e, Přepínač okrajových ploch.

a, Vstupní obrázek může být jakýkoliv 2,16,256 úrovňový (s paletou i bez), DMT nebo true color. Musí v něm existovat souvislé oblasti se shodným jasem.

b, Jméno výstupního bloku specifikuje cestu, kam budou ukládána vektorizovaná data. Na cílovém výstupním zařízení musí být dostatek místa pro uložení pomocných dat.

c, Minimální plocha je určována v pixelech (na požádání přepočítám na Ha). Je-li zadána velmi malá <1,10>, pak dojde k extrémnímu nárůstu délky vektorového obrázku.

d, Odchylka vektorové linie určuje vyhlazení linie a vyjímání přebytečných vertexů. Je zadávána v jednotce bod/bod přímky. Doporučuji hodnoty od 0.1 do 0.9. Je-li zadáno příliš malé číslo, vycházejí linie kostrbaté. Naopak při vyšší hodnotě dochází ke změnám tvaru vektorizované plochy.

e, Je-li přepínač okrajových ploch sepnut, pak jsou do výstupního vektorového obrázku posílány i plochy, které sousedí s okrajem obrazovky. V opačném případě jsou uvedené plochy ignorovány a z výsledného souboru vyloučeny.

8. Proč je výstupní soubor tak dlouhý?

Mějme šachovici o velikosti políčka 8x8 (tj 64 bodů) - tj téměř nejhorší případ. Každé políčko je ve výsledku popsáno 1 plochou. Každé políčko má 4 linie. O každou linii se dělí se sousedním políčkem.

Délka linie	=	103 byte	+ 2 krajní body 24 byte (4x6)
Délka plochy	=	71 byte	

lpl= 4(103+24) /2 + 71 = 325 byte*

lpl je počet byte potřebný pro jedno políčko šachovnice ve vektorovém formátu. V rastrové reprezentaci byla délka dané plochy 64 byte (pro 256 barev), nebo 32 byte pro 16 barev. To znamená 5 (nebo 10) násobné zvětšení objemu dat.

Vektorový formát je výrazně (co do objemu dat) úspornější až od minimální plochy 325 bodů, což odpovídá min velikosti políčka 18x18. Ovšem pro velikost plochy 1 může být velikost vektorového obrázku až **300x** větší, než původního rastru. Uživatel proto musí pečlivě vážit jaké detaily ve vektorových datech potřebuje (anebo mít dostatek diskového prostoru).

9. Závěr

V tomto článku popisovaná metoda automatické vektorizace může ušetřit mnoho práce lidem, kteří dříve museli provádět vektorizaci leteckých snímků ručně. Metoda v současné implememtaci je schopna vektorizovat i poměrně velké obrázky i několik set MB. Je však potřeba mít na disku dostatek volného protoru a seznámit se s činností parametrů a jejich vlivu na kvalitu výstupu pro danou úlohu.

Použitá literatura:

Počítačové vidění Václav Hlaváč, Milan Šoňka Grada 1992

 © December 12, 1997