Detekce hran s využitím dynamického programování

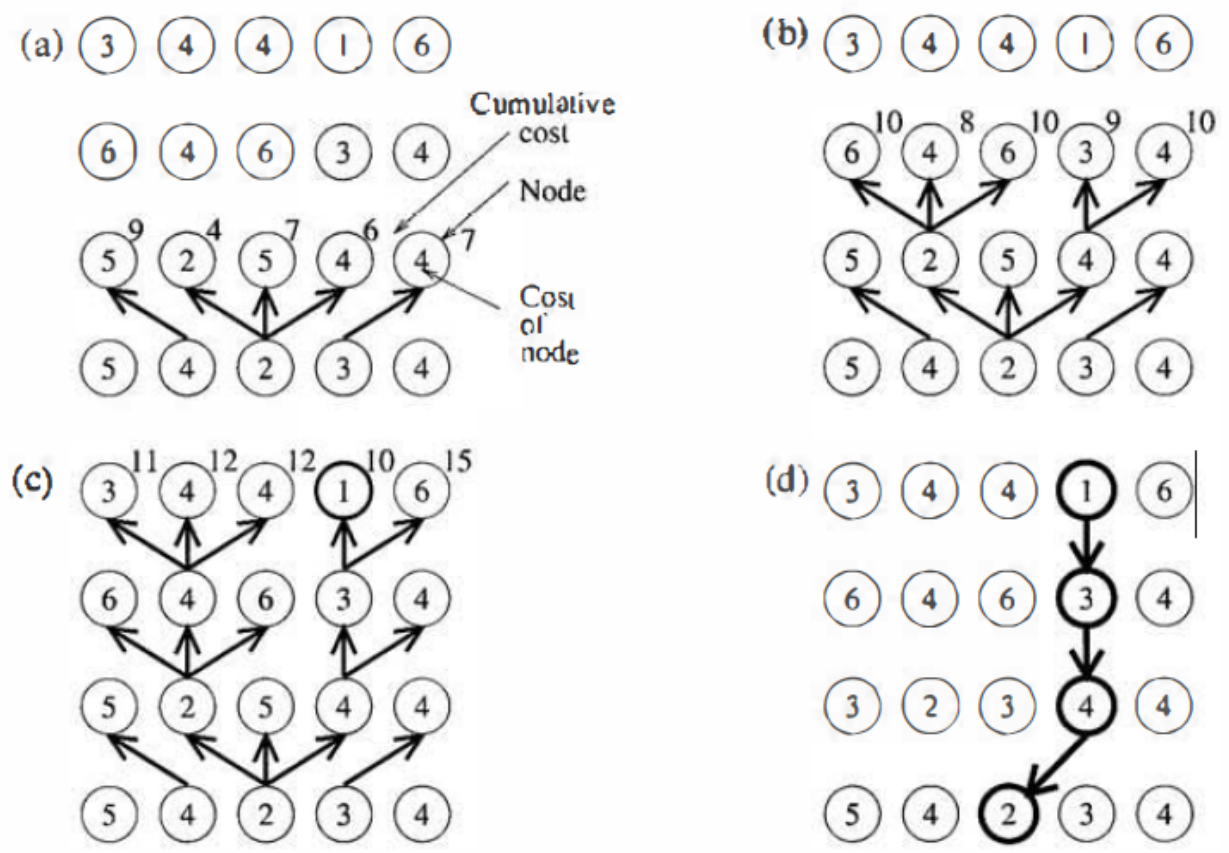
Pavel Macenauer
xmacen02@stud.fit.vutbr.cz

Jan Bureš
xbures19@stud.fit.vutbr.cz

Fakulta Informačních Technologií
Vysoké Učení Technické v Brně

9. května 2014

DP a detekce hran



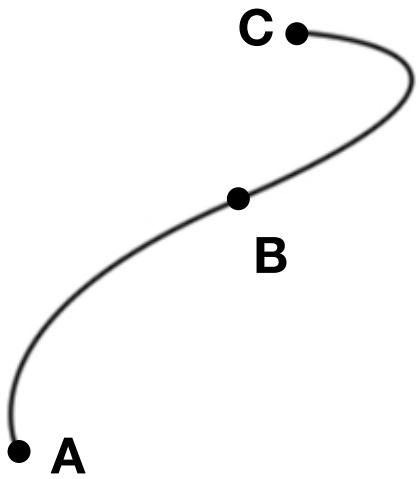
DP a detekce hran

Postup je znázorněn odzdola nahoru, ale je možné postupovat jakýmkoliv jiným směrem. Danému řádku/sloupci budeme říkat úroveň.

1. Pro každý uzel (pixel) obrazu se spočítá ohodnocení pomocí vhodné funkce
2. Pro každý uzel na dané úrovní vybereme jeho předchůdce a přičteme ohodnocení předchůdce k ohodnocení daného uzlu. Předchůdce se vybírá jako uzel s nejnižším ohodnocením ze sousedních pixelů v předchozí úrovni. Dostáváme akumulovanou cenu.
3. Na poslední úrovní vybereme nejnižší akumulovanou cenu a sledováním předchůdců dostaneme optimální cestu (hranu).

Bellmanův princip optimality

Zbývající část optimální strategie je rovněž optimální, pokud proces začíná ve stavu, do kterého se dostal v důsledku použití optimální strategie.



Obrázek : Je-li cesta z **A** do **C** optimální, pak existuje i cesta z **B** do **C**

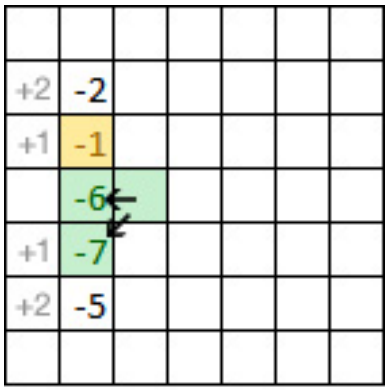
Využití a na co se to hodí

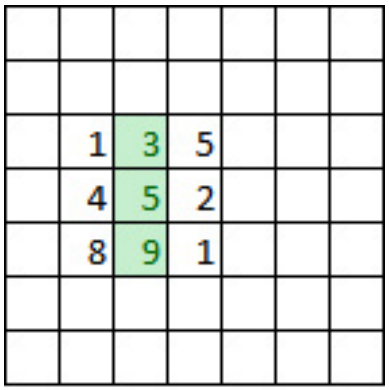
- **Hrany podlouhlých objektů** - hor, řek
- **Zdravotnictví** - cévy, tepny, páteř
- **Segmentace** - obraz je procházen od počáteční do koncové souřadnice, např. při průchodu z jednoho pixelu koncové vrstvy do pixelu první vrstvy rozdělíme obraz na 2 části podle nějaké hrany, která je cestou mezi těmito dvěma body
- **Výpočetní jednoduchost** - je třeba spočítat pouze ohodnocení všech uzlů a určit jejich předky, následně už samotný průchod je lineární

Vyhodnocování uzlů

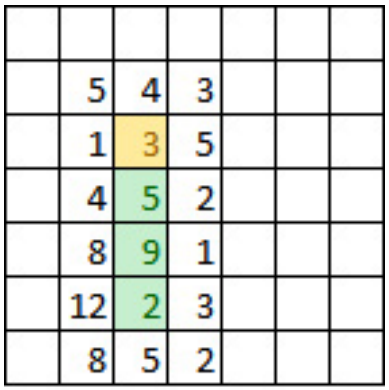
$$E_{sum} = C_{discont} E_{discont}(p_{i-1}, p_i) - C_{int} E_{int}(p_i) - C_{grad} E_{grad}(p_i)$$

E . . . jednotlivé cenové funkce
C . . . váhy pro dané funkce
p_i . . . *pixel*



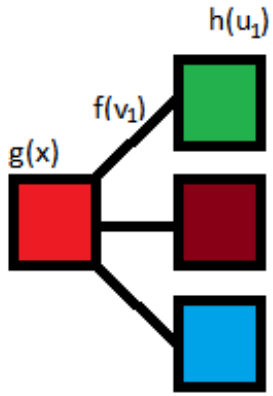


1*3 + 0*5 + (-1)*9 = $E_{grad}(p_i)$



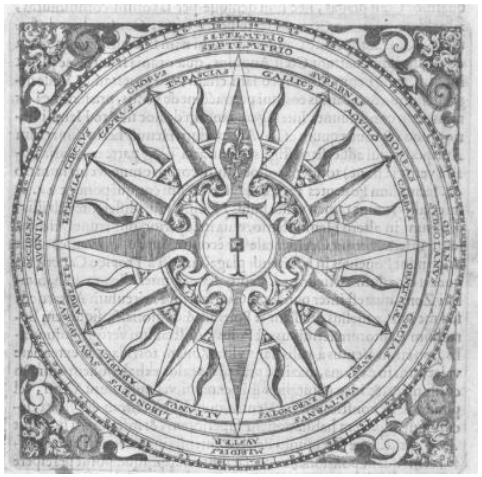
3 + 5 + 9 + 2 = 19 = $E_{int}(p_i)$

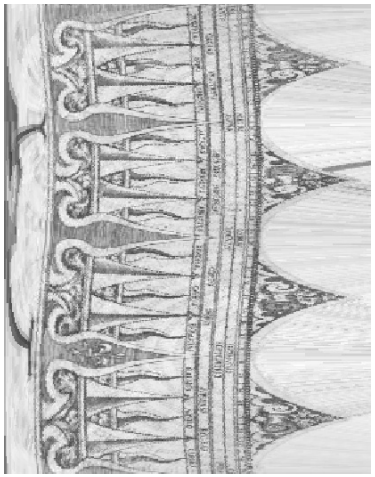
Cenové funkce



- **rozdíl RGB složek** - rozdíl jednotlivých barevných složek 2 sousedních pixelů a následný součet
- **rozdíl CMYK složek** - stejné jako RGB, pouze s převodem na CMYK
- **rozdíl odstínů šedé** - stejné jako RGB, pouze s převodem na odstíny šedé

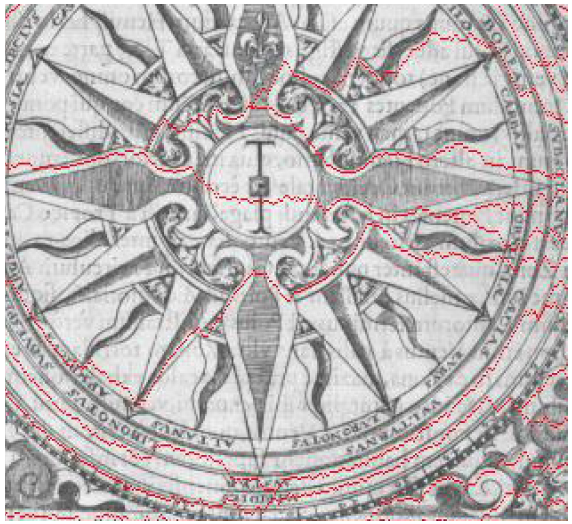
Hledání cest v polárním prostoru

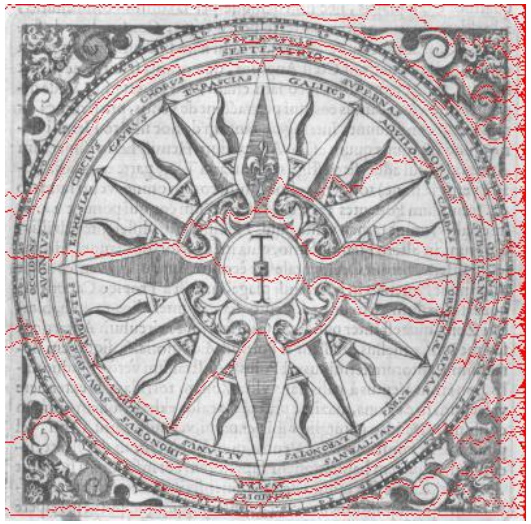


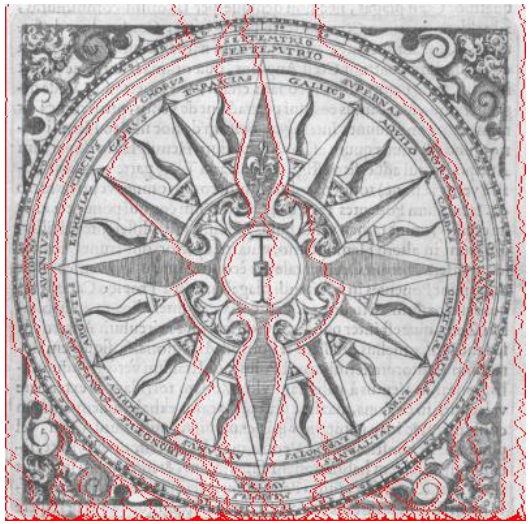


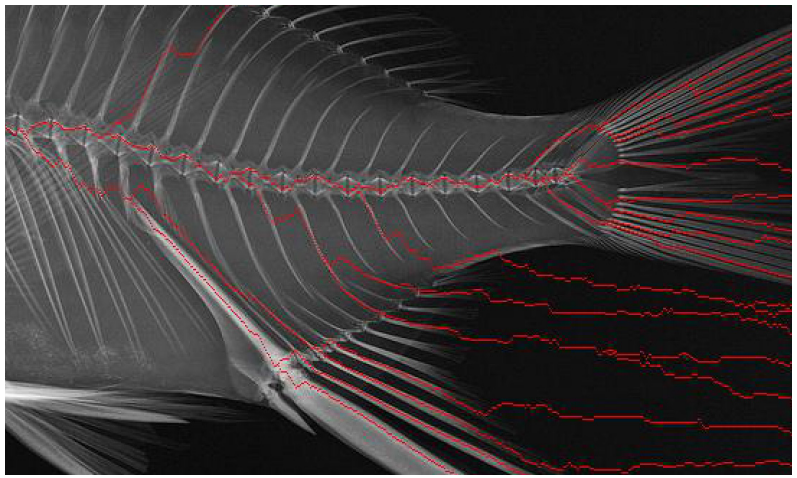
- kulaté objekty jsou v polárním prostoru lineární, je tak jednodušší detekovat hranu
- implementace pomocí lookup tabulky souřadnic
  - vytvoří se polární kopie obrazu z kartézského
  - do tabulky se zapíše jaký polární bod odpovídá jakému kartézskému
  - provede se detekce hran
  - podle tabulky se zpětně zjistí odpovídající body

Výsledky lineární detekce hran









Výsledky polární detekce hran