

Geometrická reprezentace obrazu pomocí evolučních algoritmů

Tomáš Denemark

FJFI ČVUT

16. května 2011

Evoluční algoritmy vychází z principů evoluční teorie, používají se při řešení optimalizačních úloh.

Dělí se do několika kategorií

- genetické algoritmy
- genetické a evoluční programování
- evoluční strategie

Všechny algoritmy pracují na následujícím principu:

Definuje se tzv. fitness function, která všem prvkům definičního oboru úlohy přiřazuje číselnou hodnotu a která vyjadřuje "vhodnost" prvku. Optimální řešení úlohy pak maximalizuje nebo minimalizuje fitness function.

Postup:

- 1 v definičním oboru se náhodně rozmístí určité množství náhodných jedinců
- 2 všichni nebo někteří jedinci se nahradí novými podle určených pravidel
- 3 současní jedinci se ohodnotí pomocí fitness function
- 4 pokud nějaký jedinec má požadovanou hodnotu, konec, jinak opakuj

Velmi obdobné jsou algoritmy Swarm optimization.

Existuje řada možností, jak vybírat jedince, kteří budou pozměněni

- elitistický výběr
- vážený náhodný výběr
- hierarchický výběr
- ...

Vybrané jedince můžeme upravovat dvěma způsoby

- mutací (pozměněním určitých vlastností jedince)
- rekombinací (vytvořením nového jedince kombinací vlastností několika "rodičů")

Uživatel nastaví omezení pro možné úpravy, o použitých hodnotách avšak rozhodne generátor náhodných čísel.

Pro různé úlohy může být vhodná jiná volba možných úprav.

Výhody

- obrovské úlohy
- nečekaná řešení
- odolnost na lokální extrémy fitness function
- není nutné zcela pochopit úlohu k nalezení dobrého řešení
- problém lze dobře paralelizovat

Nevýhody

- výpočetní náročnost
- obtížná volba vhodné fitness function
- téměř jistě nenalezne nejlepší možné řešení
- paralelizace není triviální

Myšlenka je inspirována prací programátora a umělce Karla Simse, který se zabýval využitím evolučních algoritmů v umění

http://www.youtube.com/watch?v=JBgG_VSP7f8

Cílem je aproximovat zdrojový obraz (bitmapu) pomocí jednoduchých vektorových útvarů. Vyrobíme tedy řadu jedinců (různých aproximací) a necháme je vyvíjet, až získáme dostatečně vhodného jedince.

První úspěšná implementace - Roger Alsing, 2008

V jeho implementaci

- soupeřil jeden potomek s jedním rodičem
- jako fitness function byla použita suma rozdílů hodnot v jednotlivých pixelech
- obraz aproximovala pomocí částečně průhledných n -úhelníků
- v současné verzi používá křivky splines a aproximace obrázku 200x200 trvá přibližně 90s

Na co je důležité dávat pozor

- jednoduché fitness functions ignorují detaily, vhodná volba je kvadrát vzdálenosti
- systém barev RGB špatně postihuje vlastnosti lidského oka
- obzvláště fotografie obsahují barevné přechody, které je nutno aproximovat velkým množstvím útvarů nebo útvary s nejasnými kraji nebo útvary s přechodovou výplní
- pokud se algoritmu nedá dostatečná volnost, bude hledat alternativní řešení, která téměř vždy lidskému oku připadají jako artefakty
- fitness function nemusí být konstantní v čase, s počtem generací se může měnit
- preprocessing a postprocessing mohou výrazně pomoci

Zajímavé implementace

- Roger Alsing - EvoLisa, první projekt svého druhu
<http://rogeralsing.com/2008/12/07/genetic-programming-evolution-of-mona-lisa/>
- Screaming Duck Software - výborné výsledky, časově náročné
<http://www.screamingduck.com/Article.php?ArticleID=46&Show=ABCE>
- Mario Klingermann - komprese do 140 znaků
<http://www.flickr.com/photos/quasimondo/3518306770/>
- Matt Ditton - použití mesh-grid
<http://vimeo.com/7052067>
- VecGen - využívá kruhy
<http://sourceforge.net/projects/vecgen/>
- Chris Sloetjes - komprese videa
<http://code.google.com/p/clam-compression-studio/>