Spracovanie obrazu v reálnom čase, softvér (1)

Cieľom článku je zhrnúť metódy spracovania obrazu, ktoré by sa dali využiť pri spracovaní v reálnom čase. Článok je zameraný na softvérovú časť spracovania obrazu a nadväzuje na článok z AT&P 12/2008 „Spracovanie obrazu v reálnom čase, hardvér".

1. Algoritmy spracovania obrazu a ich zjednodušenie

Čo je to algoritmus? Je to zoznam predpísaných pravidiel alebo procedúr na riešenie problému. Na spracovanie obrazu existuje veľa algoritmov, avšak pri uvažovaní spracovania obrazu v reálnom čase treba hľadať efektívne algoritmy.

Efektivita naznačuje nízku výpočtovú a pamäťovú náročnosť. Pre veľké množstvo vstupných údajov pri spracovaní obrazu je vývoj algoritmov ťažkou úlohou. Pre lepšie vlastnosti sú algoritmy často optimalizované pre danú hardvérovú platformu. Ďalšou možnosťou, ako zrýchliť spracovanie, je získanie výsledku prostredníctvom nižšej algoritmickej úrovne; takáto modifikácia algoritmu pomáha znížiť hardvérové nároky pri spracovaní. Táto kapitola určuje stratégiu implementácie algoritmov na využitie v reálnom čase.

Pri skúmaní spracovania obrazu v reálnom čase by sme mohli zjednodušovanie spracovania obrazu rozdeliť takto:
- zníženie počtu operácií,
- redukcia množstva údajov,
- použitie jednoduchších alebo zjednodušených algoritmov.

Najpoužívanejšími stratégiami zjednodušovania sú redukcia množstva údajov a zníženie počtu operácií.

1.1 Zníženie počtu operácií

V nízkej úrovni spracovania obrazu je každá operácia obzvlášť náročná na čas, pretože ide o veľké množstvo údajov. Preto hrá zníženie počtu operácií hlavnú úlohu pri zjednodušovaní spracovania obrazu.

1.1.1 Čisté zníženie počtu operácií

Stratégia čistého zníženia počtu operácií spočíva v tom, že sa zredukuje počet operácií, ktoré sú potrebné na dosiahnutie výsledku, avšak výsledok zostane nezmenený. Ak sa na základe redukcie zmení aj výsledok, hovoríme o aproximácii alebo o alternatívnom riešení; tieto riešenia si opíšeme neskôr. Kandidátmi na takéto operácie sú tie operátory, ktoré majú symetriu alebo nadbytočnosť v počítaní. Aplikácia tejto stratégie spočíva v odkrývaní skrytej súmernosti alebo nadbytku v počítaní, ktoré môžu byť často objavené až pri ručnom počítaní a pozornom zaznamenávaní si matematických totožností alebo vlastností. Napríklad filter, ktorý potrebuje pri výpočte výstupu aj okolité obrazové elementy, ktoré sa navzájom sčítajú. Posúvaním okienka (napr. 3 x 3) po obraze a opätovným sčítavaním ide o nadbytočné sčítavanie a teda po optimalizácii by sa postupovalo nasledovne. Po výpočte by sa výsledný súčet nechával v pamäti a pripočítal by sa ďalší stĺpec (sprava) a odpočítal (zľava). Zníženie počítania môže byť dosiahnuté aj preskupením takých operácií, ktoré sú časovo náročné a medzi ktoré patria aj násobenie a delenie. Príklad a * b + a * c môžeme zjednodušiť na (b + c)*a. Operácie, ako delenie a násobenie, sú považované za časovo náročné, lebo potrebujú väčší výpočtový výkon, a preto sa občas nahradzujú aj bitovým posunom. Používanie výpočtovo jednoduchých operácií sa považuje za jeden z kľúčových krokov k dosiahnutiu rýchlejšieho spracovania obrazu.

1.1.2 Aproximačné zníženie počtu operácií

Aproximačná stratégia sa líši od čistej tým, že výpočtové operácie sa nahradia novými jednoduchšími operáciami, avšak výsledok nebude ten istý, ale približný. Cieľom je minimalizovať chyby (ako je to len možné) počas prijateľného časového intervalu určeného na spracovanie. Ide o hľadanie kompromisu medzi presnosťou a rýchlosťou.

1.1.3 Alternatívne zníženie počtu operácií

Alternatívna stratégia sa používa na dosiahnutie rýchlejšieho spracovania obrazu a zároveň udržuje úroveň presnosti. Táto stratégia je podobná ako aproximačná až na to, že sa v prvom rade používa tam, kde ide o zložitý výpočet pre aproximáciu alebo kde sa nedá realizovať zmenšenie objemu vstupných údajov.

1.2 Redukcia množstva údajov

Táto stratégia transformuje vstupné údaje s cieľom zmenšenia množstva údajov, ktoré následne zrýchlia spracovanie údajov. Zmenšovanie množstva údajov pri spracovaní obrazu má mnoho foriem – rozdeľovanie, regióny záujmu, výberové spracovanie atď. Vo všetkých prípadoch je však výstupná podmnožina tvorená z množiny obrazových elementov, ktoré sú na vstupe tejto operácie.
- **Zníženie rozlíšenia** – na zníženie rozlíšenia je viacero metód realizácie. Poznamenajme však, že nie pre všetky aplikácie je tento postup vhodný.
- **Rozdeľovanie obrazu do blokov** – je spôsob, keď sa rozdelí vstupný obraz na bloky, ktoré sa neprekrývajú a následne sa spracúvajú oddelene.
- **Oblasť záujmu** – zo vstupného obrazu je vybraná len tá časť, o ktorú máme záujem, a ďalej sa spracúva len táto oblasť.
- **Viac rozlíšení** – je založené na spôsobe, keď je hrubý odhad realizovaný s menším rozlíšením a na spresnenie výsledkov používame väčšie rozlíšenie.

1.3 Zvýraznenie vhodných rysov

Výber vhodných rysov objektu tiež pomáha znížiť množstvo informácií potrebných pri vyhodnotení obrazu. Vhodné rysy závisia od aplikácií a predmetov potrebných na rozpoznanie či riadenie. Ak sú nájdené vhodné rysy, môže sa pre ne vytvoriť prostriedok na ich efektívne zvýraznenie. Prostriedok, ktorý si nastavuje parametre sám, je výhodnejší ako ten, ktorý potrebuje nastaviť parametre manuálne, lebo v prípade komplikácií sa samonastavujúci prostriedok dokáže rýchlejšie prispôsobiť situácii.

1.4 Zjednodušené algoritmy

Vo všeobecnosti základným pravidlom pri spracovaní obrazu je použitie jednoduchých alebo zjednodušených algoritmov. Pravidlo pre spracovanie obrazu v reálnom čase je aplikovať jednoduché operácie. To znamená vyhľadávať jednoduché riešenia s používaním jednoduchých operácií. Avšak nie každá matematicky jednoduchá operácia je vhodná aj na spracovanie obrazu v reálnom čase. Treba zohľadniť aj požiadavky na ukladanie údajov a nielen na výpočtovú náročnosť.

2. Softvérové riešenia spracovania obrazu

Softvérové metódy tvoria ďalší aspekt pri spracovaní obrazu. Dosť často algoritmy, ktoré dosahujú uspokojivý výkon vo vývojárskom prostredí, nedokážu zrealizovať rovnaký výkon na cieľovej hardvérovej platforme bez akýchkoľvek úprav. Tento efekt sa viac prejavuje na prenosných zariadeniach s obmedzenou frekvenciou a obmedzeným množstvom pamäte, ale aj na moderných pracovných staniciach s výkonným procesorom (GPP). Kvôli tomuto obmedzeniu musia byť algoritmy upravené, aby sa mohli používať na danom hardvéri, a tak dosahovali efektívne a účinné využitie hardvéru. Zaujímať sa budeme o softvérové návrhové vzory, stránkovanie pamäte a softvérovú optimalizáciu. V predchádzajúcich kapitolách sme sa venovali algoritmom a

hardvéru potrebným na spracovanie obrazu, ďalej sa budeme zaoberať tiež rovnako dôležitou častou a to je ich implementácia pre reálny čas. Softvér je „medzivrstvou“ medzi hardvérom a algoritmami. Preto je na spracovanie obrazu v reálnom čase potrebný vývoj účinného softvéru, ktorý maximálne využije dostupný hardvér.

2.1 Elementy softvérových platforiem

Tak, ako je možný výber z veľkého množstva hardvérových platforiem na spracovanie obrazu v reálnom čase, je aj mnoho možností pre softvérovú platformu. Kľúčovými komponentmi na spracovanie obrazu sú programovací jazyk, princípy softvérovej architektúry a operačný systém reálneho času.

2.1.1 Programovacie jazyky

V prípade projektovania aplikácie na spracovanie obrazu používame dva druhy programovacích jazykov. Tie, ktoré sa používajú na vytváranie prototypu a vývoj, a tie, ktoré sa používajú na umiestnenie v nezávislých – samostatných produktoch. Jeden z problémov pri návrhu aplikácie v reálnom čase je prechod kódu z vývojového prostredia, ako je MATLAB alebo Lab-VIEW, do zdrojového kódu, ako je napríklad C alebo C++.

Programovacie jazyky určené pre výskum

Programovacie jazyky a štýl programovania používaný pri výrobe aplikácií na spracovanie obrazu v reálnom čase určených do procesu riadenia je odlišný od toho, čo sa používa počas návrhu a výskumu. Najbežnejšie vývojové jazyky používané pri výskume zahrnujú nejakú kombináciu MATLAB, LabVIEW alebo C/C++. MATLAB je interpretovaný na vysokej úrovni programovania. To znamená, že v MATLAB-e „.m“ súbore zdrojového kódu je každá inštrukcia, príkaz alebo premenná, ktoré musia byť interpretované počas behu programu. Iným programovacím rysom je, že v MATLAB-e každá premenná môže byť definovaná bez špecifikácie dátového typu. Samozrejme hlavnou výhodou MATLAB-u sú maticovo vektorové možnosti spracovania, ako aj možnosť operácií lineárnej algebry na maticových dátových štruktúrach bez použitia cyklov. MATLAB môže byť vybavený aj viacerými knižnicami s rôznymi nástrojovými setmi. Niektoré z nástrojových setov zahrnujú spracovanie obrazu alebo spracovanie signálov. MATLAB je z prevažnej časti textové programovacie prostredie, ktoré používa funkcie napísané v „.m“ súboroch, na rozdiel od Simulinku, ktorý je doplnkom a umožňuje grafické programovanie založené na blokoch. Blok Simulinku na spracovanie obrazu a videa následne uľahčuje vývoj systémov na spracovanie obrazu v porovnaní s textovým programovaním. MATLAB umožňuje využívať aj vonkajšie vzorkovanie („hardware in the loop“), čo umožňuje, že modely môžu bežať na koncovej hardvérovej platforme a údaje prechádzajú z MATLAB-u na platformu; to umožňuje rýchle ladenie aj počas chodu. Hore uvedené rysy robia z MATLAB-u silné a flexibilné programovacie prostredie umožňujúce rýchle vytváranie prototypov na spracovanie obrazu alebo videa. MATLAB poskytuje bohaté textové programovanie, kde možno rýchlo naprogramovať algoritmy, a tak overiť ich funkčnosť. Avšak rysy umožňujúce rýchly vývoj bránia algoritmom v ich implementácii na využitie v reálnom čase. Preklad inštrukcií za chodu je charakteristický pre MATLAB. V skutočnosti to má aj nevýhody a jednou takou je spomalenie vykonávania „.m“ súborov, čo sa najviac prejavuje v slučkách. Väčšina algoritmov určených na spracovanie videa a obrazu má viacnásobné slučky, a preto je MATLAB nevhodný na implementáciu do procesov reálneho času. Ak by sa vylúčili slučky a využili schopnosti maticového a vektorového počítania, čo MATLAB umožňuje, mohol by nastať problém pri prevode do ďalšieho univerzálneho jazyka, ako je napríklad C/C++. Treba poznamenať, že MATLAB ukladá obrazové údaje v stĺpcoch a väčšina programov vytvorených na C báze údaje uchovávajú v riadku. Táto skutočnosť obmedzuje použitie MATLAB s C. MATLAB nepotrebuje presné zadefinovanie dátového typu premennej, čo programy založené na C baze potrebujú. Existujú však aj nástroje na prechod z MATLAB-u do C [11].

Na rozdiel od MATLAB-u, ktorý bol navrhnutý od textovo orientovaného prostredia k Simulinku, je LabVIEW navrhnutý ako grafické programovacie prostredie. LabVIEW poskytuje silný nástroj založený na vývojom prostredí. V kombinácii nástrojov, ako je IMAQ Vision a pokročilé spracovanie signálu, môže byť LabVIEW využitý na rýchly návrh prototypu spracovania obrazu. Sila LabVIEW spočíva v grafickom programovaní, ktoré dovoľuje hierarchické a modulárne návrhy systémov cez tzv. virtuálne nástroje (VIs) a sub VIs. LabVIEW môže tiež využívať algoritmy spracovania obrazu napísané v univerzálnom programovacom jazyku, ako je C/C++ s riadkovým ukladaním údajov. Čo pomáha prechodu do textového zdrojového kódu. Výhodou LabVIEW je, že má grafické prostredie, s ktorým sa dajú ľahko nastavovať parametre v simulácii obrazu a video spracovaní a zároveň možno skontrolovať medzivýsledky.

Ak je programované spracovanie obrazu od začiatku v C/C++, je výhodou, že sa nemusí prekladať zdrojový kód pre vyšší univerzálny programovací jazyk, ako je to v prípade LabVIEW alebo MATLAB-u. Problémom s programovaním v C/C++ je stratenie výhod programovacieho jazyka, ako je MATLAB alebo LabVIEW, ktoré pomáhajú pri rýchlom vývoji algoritmov. Preto sa v praxi hľadá akákoľvek kombinácia MATLAB, LabVIEW a C/C++ pre vývoj spracovania obrazu alebo videa. Zatiaľ čo MATLAB a LabVIEW sú odolné nástroje pre výskum a vývoj, nie sú vhodné na využívanie v reálnom čase.

Literatúra bude uvedená v budúcej časti.

Pokračovanie v budúcom čísle.

Ing. Tomáš Surovčík
prof. Ing. Ladislav Jurišica, PhD.

Slovenská technická univerzita v Bratislave
Fakulta elektrotechniky a informatiky
Ústav riadenia a priemyselnej informatiky
Ilkovičova 3, 812 19 Bratislava
e-mail: tomas.surovcik@stu.sk