Technická zpráva

ÚTIA Akademie věd České republiky
Ústav teorie informace a automatizace AV ČR, v.v.i.

RIPAC Frontend
Metodologie tvorby bloků v Simulinku

Bohumil Kovář
kovar@utia.cas.cz, +420-2-6605 2502

Rapid prototyping tools for development of HW-accelerated embedded image- and video-processing applications
1ET400750408

Department of Signal Processing
http://sp.utia.cz

Obsah

Revize

Revize	Datum	Autor	Popis změn v dokumentu
0	10.12.2008	B.K.	Vytvoření dokumentu
1			
2			
3			

1 Úvod

Hlavním cílem projektu RIPAC je vytvoření metodologie a základních nástrojů pro rychlý návrh a hardwarovou implementaci (FPGA, DSP) rekonfigurovatelných systémů zpracování obrazu a videa. Předpokládané využití výsledků projektu je v průmyslových aplikacích zpracování obrazu, které jsou výpočetně náročné, případně u nichž je požadováno zpracování vstupních dat v reálném čase.

Základem systému je množina sofwarových implementací základních operací zpracování obrazu a jim ekvivalentní hardwarové implementace pro DSP a FPGA. Softwarové implementace jsou používány ve fázi návrhu a testování algoritmů v prostředí Simulinku. Tento dokument popisuje metodologii tvorby nových Simulinkových bloků tak, aby byly datově i funkčně kompatibilní s návrhy v projektu RIPAC.

2 Požadavky na systém

Jako frontend systému je používán software Matlab s nadstavbou Simulink. Pro zjednodušení vývoje a zajištění plné kompatibility s produkty a toolboxy firmy Mathworks Inc. byl navíc použit:

- Image Acquisition Toolbox,
- Image Processing Toolbox,
- Video and Image Processing Blockset.

Uvedené nástroje jsou používány zejména pro I/O operace. Místo ostatních funkčních bloků těchto toolboxů je doporučeno používat vlastní implementace, tak aby byla zajištěna plná kompatibilita mezi simulační a hardwarovou částí systému RIPAC.

Systém je navržen tak, aby bylo možné, bez ztráty funkčnosti, nahradit Matlab-Simulink frontend jako celek jiným, licenčně a ekonomicky méně náročným řešením (OpenSource).

3 Funkční bloky

V této části je popsána tvorba uživatelských simulinkových bloků, kompatibilních se systémem RIPAC. Pro vstup z digitálních kamer se předpokládá použití *Image Acquisition Toolboxu*, při vstupu ze souboru (obraz, video) pak *Video and Image Processing Blocksetu*. Pro výstup na obrazovku, případně do souboru, pak opět *Video and Image Processing Blocksetu*. Předpokládá se, že vstupní data jsou monochromatická 8 bitová. Použitý datový typ tedy je **uint8**. Ten musí být zachován na vstupu i výstupu všech uživatelských bloků. Při propojení uživatelských bloků s bloky z nadstaveb Matlabu, které používají jiné datové typy, je nutné zajistit potřebné datové konverze.

3.1 Metodologie

Uživatelské bloky jsou v Simulinku implementovány zejména jako uživatelské funkce. Doporučeno je používat Level-2 M-File S-Funkce. To umožňuje velmi rychlý převod algoritmů implementovaných v Matlabu jako m-file (funkce nebo script) do prostředí Simulinku. Pouze výpočetně náročné bloky, které není možné dekomponovat na posloupnost výpočetně jednodušších bloků, je vhodné implementovat jako C/C++ S-Funkce. Level-2 M-file S-function API umožňuje vytvářet bloky, které mají většinu vlastností a možností vestavěných bloků systému Simulink. Například:

- více vstupních i výstupních portů (MIMO),
- 1-D, 2-D, a n-D vstupní a výstupní signály

- datové typy podporované v Simulinku,
- uživatelsky definované datové typy,
- run-time parametry.

Uživatelsky definovaná Level-2 M-File-2 S-Funkce (dále jen msf) se skládá ze tří částí. V první části jsou definovány parametry bloku (počet vstupů a výstupů, datové typy, vzorkování, atp.). Ve druhé části je definována vlastní metoda (prováděná funkce) bloku a pokud je to nutné, načtení parametrů bloku. Třetí část obsahuje vlastní implementaci zapouzdřené funkce. Tím, že je oddělena implementace funkce od parametrů bloku, je dosaženo snadné přenositelnosti do jiných programovacích jazyků a systémů. V projektu RIPAC bylo této vlastnosti využito při přípravě VHDL kódu v programu SynplifyDSP (nadstavba Simulinku).

Jednoduchá funkce tedu může vypadat například takto:

1. deklarace funkce a inicializace bloku

```
function msfThreshold(block)
%% Level-2 M-File S-Function for thresholding images
%% RIPAC Blockset
%%=====================================================
setup(block);
```

2. nastavení parametrů bloku

```
function setup(block)

    % Register number of ports
    block.NumInputPorts  = 1;
    block.NumOutputPorts = 1;

    % Setup port properties to be inherited or dynamic
    block.SetPreCompInpPortInfoToDynamic;
    block.SetPreCompOutPortInfoToDynamic;

    % Override input port properties
    block.InputPort(1).DatatypeID  = 3; % uint8
    block.InputPort(1).Complexity  = 'Real';
    block.InputPort(1).SamplingMode = 'Sample';
    block.InputPort(1).Overwritable = false; % No in-place operation

    % Override output port properties
    block.OutputPort(1).DatatypeID  = 3; % uint8
    block.OutputPort(1).Complexity  = 'Real';
    block.OutputPort(1).SamplingMode = 'Sample';

    % Register parameters
    block.NumDialogPrms     = 1;
    block.DialogPrmsTunable = {'Tunable'};

    % Register methods
    block.RegBlockMethod('Outputs',@Output);
```

3. Implementace metody bloku

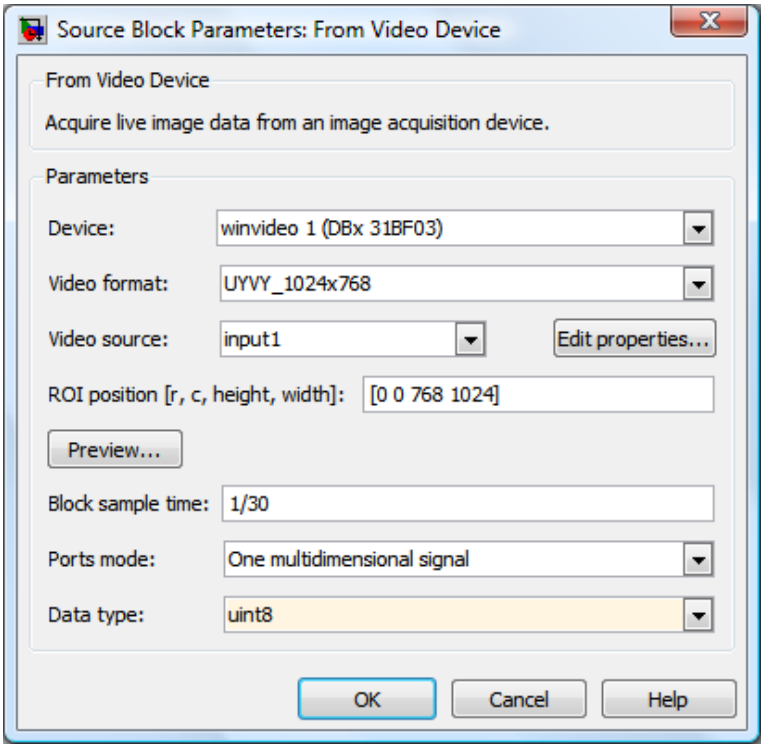
```
function Output(block)
    thr = block.DialogPrm(1).Data;
    block.OutputPort(1).Data = Threshold(block.InputPort(1).Data,thr);
```

4. Implementace vlastní uživatelské funkce

```
function g = Threshold(f,thr)
    g = f;
    f = f(:,:) >= thr;
    g = uint8(f) .* g;
```

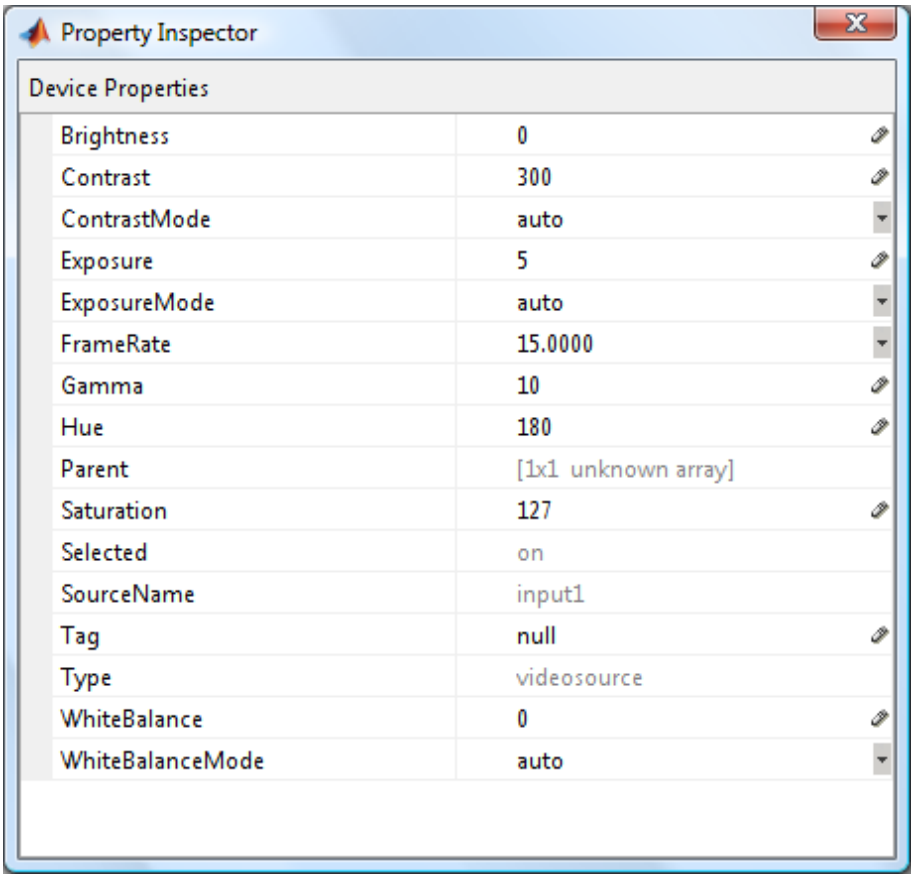
3.2 Vstupy a výstupy

V projektu RIPAC předpokládáme, že vstupní obrazová data jsou snímána buď digitální FireWire (IEEE 1394) kamerou ImagingSource DBK 31BF03 nebo jsou uložena v souborech jako videosekvence, případně fotografie v běžných grafických formátech. Při práci s kamerou je použitý *Image Acquisition Toolbox*. Tento nástroj vytváří interface mezi Matlabem a kamerou a umožňuje nastavovat parametry a řídit zachytávání obrazu z kamery. Použití Image Acquisition Toolboxu je podmíněno platnou licencí nástroje Image Processing Toolbox (i když tato nadstavba Matlabu není používána).



Obrázek 1: Nastavení video zařízení

V operačním systému Linux není Image Acquisition Toolbox ve verzi 3.0. (Matlab2008b) zatím podporován. V Linuxu tedy musí být obrazová data zaznamenána do souboru a do Simulinku načtena například *Video and Image Processing Blocksetem*. Pro výstup dat na obrazovku monitoru, případně do souboru, jsou v obou hlavních operačních systémech použity funkční bloky z Video and Image Processing Blocksetu.



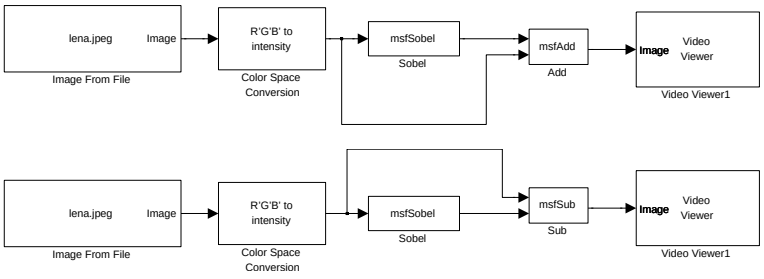
Obrázek 2: Parametry snímání obrazu

3.3 Příklady implementací

V této kapitole je uvedeno několik vzorových implementací uživatelských bloků. Popsána je funkce bloků, vlastní implementace, použití v Simulinku a vlastní výstupy. Zdrojové kódy jsou na přiloženém CD-ROM. Jako vstupní obraz byl vždy použit barevný obrázek Lena o rozlišení 512x512, který byl z důvodu zachování datové kompatibility v systému RIPAC převeden na 8 bitový monochromatický obraz.

Obrázek 3: Testovací obrázky

1. Add a **Sub**
Součet a rozdíl dvou vstupních obrazů.



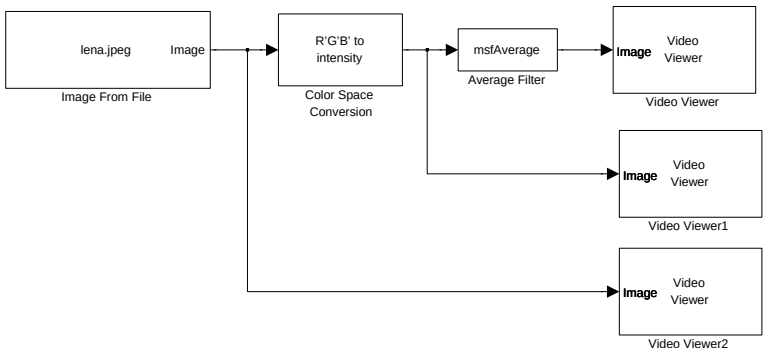
(a)

(b)

Obrázek 4: Součet (a) a rozdíl (b) vstupního obrazu a detekovaných hran.

3. Average
Průměrování v lokálním okolí. Obyčejné průměrování filtruje obraz tím, že jako nový jas bodu přiřadí aritmetický (nebo vážený aritmetický) průměr jasu bodů obdelníkového lokálního okolí. Simulinkový blok Average má implementována 3 základní konvoluční jádra.

Parametr Funkce	Konvoluční jádro
1	$h_1 = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$
2	$h_2 = \frac{1}{10}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
3	$h_3 = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$




Obrázek 5: Vyhlazení obrazu průměrováním. Použito konvoluční jádro (3).

4. Sobel

Sobelův hranový detektor je vhodnou aproximací digitálního gradientu, kterým jsou aproximovány první parciální derivcace. Proto je směrově závislý. V bloku sobel jsou implementovány dva základní směry. Ostatní orientace gradientu (parametr funkce 3) jsou z nich dopočítávány.

Parametr Funkce	Konvoluční jádro	Orientace
1	$h_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$	horizontální
2	$h_2 = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$	vertikální

Implementace:

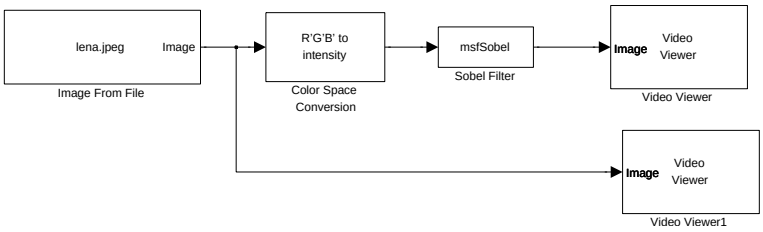
```
function  g = sobel_filter(f, dir)

  %% Sobel filter coefficients
  h = [1 2 1; 0 0 0; -1 -2 -1];

  if dir == 1 % Vertial
    g = abs(filter2(h,f));
  elseif dir == 2 % Horizontal
    g = abs(filter2(h',f));
  else % All
    g = abs(filter2(h, f)) + abs(filter2(h',f));
  end

  %% Prevent overflow when converting to uint8
  g(g>255) = 255;

  g = uint8(round(g));
```



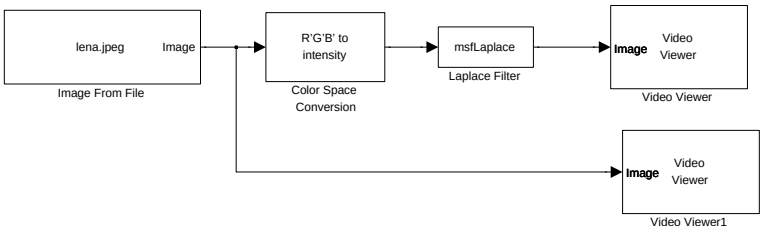
Obrázek 6: Sobelův hranový detektor

5. Laplace

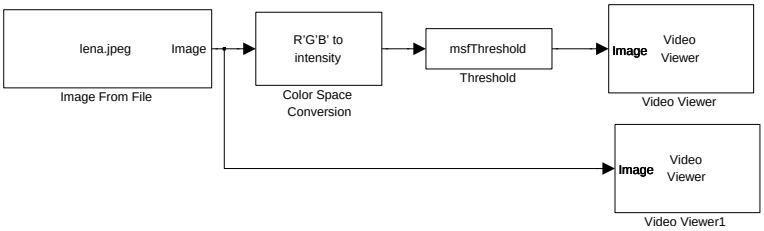
Laplaceův hranový detektor je velmi populární gradientní operátor, který aproximuje 2.derivaci obrazové funkce. Proto je invariantní vůči otočení a udává pouze velikost hrany a nikoliv její směr. Implementována jsou dvě nejpoužívanější konvoluční jádra.

Parametr Funkce	Konvoluční jádro
1	$h_1 = \frac{1}{5}\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$
2	$h_2 = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

Obrázek 7: Aproximace 2. derivace obrazové funkce - Laplaceův operátor



6. Threshold
Prahování obrazu s jedním prahem.



Obrázek 8: Prahování obrazu.

4 Instalace

Uvedený software není potřeba instalovat. Stačí nakopírovat adresář SRC z přiloženého CD na lokální disk a v Matlabu přidat cestu (addpath) k tomuto adresáři, případně ho pouze nastavit jako aktivní (cd). Pak je možné otevřít a spouštět ukázkové simulinkové modely, případně přidávat ukázkové Simulink Level-2 M-File S-Funkce do vlastních modelů.

5 Závěr

V tomto dokumentu je popsána metodologie tvorby uživatelských Simulink funkcí kompatibilních se systémem RIPAC. V případě rekonfigurovatelných algoritmů zpracování videa a obrazu je nutné, aby vlastní algoritmus byl dekomponován na co možná nejjednodužší (elementární) operace, které je možné v průběhu zpracování úlohy znovu a bez překladu použít. Tento přístup značně urychluje vývoj a hardwarové implementace průmyslových aplikací zpracování videa a obrazu.

Tato zpráva vznikla za podpory grantu AV ČR č. 1ET400750408 – Rapid prototyping tools for development of HW-accelerated embedded image- and video-processing applications.

6 Výpis CDROM

```
CDROM:

    DOC/
         SBlocks.pdf
    IMG/
         lena.jpg
         lena512.bmp
    SRC/
         msfAdd.m
         msfAverage.m
         msfLaplace.m
         msfSobel.m
         msfSub.m
         msfThreshold.m
         rAdd.mdl
         rAverage.mdl
         rLaplace.mdl
         rSobel.mdl
         rSub.mdl
         rThreshold.mdl
         template.mdl
```