

2:1 Reduction / expansion

Obsah

1. Popis.....	1
2. Závislosti	1
3. Implementačné informácie	1
4. Piny	2
5. Konfigurácia	2
5.1 Konfiguračná štruktúra.....	2
5.2 Konfiguračný dialóg.....	3
6. Definícia triedy	3
7. Príklad použitia	4

1. Popis



Trieda *Reduction* zredukuje rozlíšenie obrazu na polovicu v horizontálnom, vertikálnom alebo oboch smeroch. Redukcia sa realizuje priemerovaním. Tak isto je možné vykonať opačnú operáciu – expanziu rozlíšenia na dvojnásobok. Expanzia sa realizuje interpoláciou chýbajúcich hodnôt.

2. Závislosti

Trieda *Reduction* používa tieto externé definíčné súbory a knižnice:

Headers: *filtergraph.h*, *datatypes.h*

Libs: *filtergraph.lib*

3. Implementačné informácie

Informácie o triede:

Názov: *Reduction*

Verzia: *1.0*

Magic: *100*

Informácie o definíciách:

Header: *reduction.h*

Konfiguračný header: *reductioncfg.h*

Lib: *reduction.lib*

Informácie o knižnici obsahujúcej triedu:

Názov knižnice: *Reduction Library*

Verzia knižnice: 1.0

Dll súbor knižnice: *reduction.dll*

4. Piny

Piny sú popísané spôsobom: „*[index pinu na filtri] názov pinu: popis pinu*“.

[0] Cfg in: Konfiguračný vstup. Vstupom je serializovaná konfigurácia.

[1] Cfg out: Konfiguračný výstup. Výstupom je serializovaná konfigurácia.

[2] In: Vstup. Vstupom je šedo tónová rovina v štruktúre *TPlane*.

[3] Out: Výstup. Výstupom je zredukovaná alebo expandovaná šedo tónová rovina v štruktúre *TPlane*.

5. Konfigurácia

5.1 Konfiguračná štruktúra

Konfiguračná štruktúra je definovaná nasledovne:

```
typedef struct {  
    int      Action;           // akcia  
    bool     HCorrection;      // horizontálna korekcia  
    bool     VCorrection;      // vertikálna korekcia  
} TReductionConfig;  
  
// akcie  
RED_HORIZONTAL      0      // horizontálna redukcia  
RED_VERTICAL        1      // vertikálna redukcia  
RED_HV              2      // redukcia v oboch smeroch  
EXP_HORIZONTAL      3      // horizontálna expanzia  
EXP_VERTICAL        4      // vertikálna expanzia  
EXP_HV              5      // expanzia v oboch smeroch
```

Parameter *Action* určuje operáciu, ktorá sa bude vykonávať.

Ak vstupný obraz mal rozlíšenie $C_x \times C_y$, tak jednotlivé akcie ho zmenia nasledovne (lomeno znamená celočíselné delenie):

horizontálna redukcia : $C_x/2 \times C_y$

vertikálnej redukcia: $C_x \times C_y/2$

redukcii v oboch smeroch: $C_x/2 \times C_y/2$

horizontálna expanzia: $2C_x \times C_y$

vertikálna expanzia: $C_x \times 2C_y$

expanzia v oboch smeroch: $2C_x \times 2C_y$.

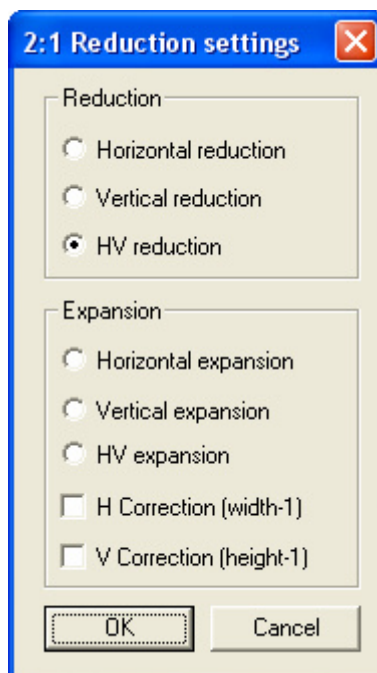
V prípade redukovania obrazu z nepárny rozlíšením, je daný rozmer po redukcii ešte zväčšený o 1, kvôli zachovaniu posledného riadku alebo stĺpca, ktorý nemal byť s čím priemerovaný a bol by úplne stratený.

V prípade potreby expandovať obraz do nepárneho rozlíšenia je možné zapnúť vybrané korekcie. Horizontálna korekcia zmenší rozlíšenie o 1 v horizontálnom smere, teda výsledné bude C_x-1 , podobne vertikálna korekcia zmenší vertikálne rozlíšenie o 1, teda na C_y-1 .

Príklad s nepárny rozlíšením:

Obraz 513x513 -> HV redukcia -> 257x257 -> HV expanzia -> 514x514, ale 257x257 -> HV expanzia s H a V korekciou -> 513x513.

5.2 Konfiguračný dialóg



Konfiguračný dialóg poskytuje voľby popísané pri konfiguračnej štruktúre. V bloku *Reduction* si môžeme zvoliť typ redukcie. V bloku *Expansion* si môžeme zvoliť typ expanzie a zapnúť potrebné korekcie.

6. Definícia triedy

Trieda *Reduction* je definovaná nasledovne:

```
class TReductionFilter : public TFilter {
private:
    HINSTANCE hInstance;           // identifikácia inštancie knižnice

    int        Action;             // konfigurácia...
    bool       HCorrection;
    bool       VCorrection;

    int Width;                    // šírka výstupného obrazu
    int Height;                   // výška výstupného obrazu

    TPlane* InPlane;              // vstupná rovina
    TPlane OutPlane;              // výstupná rovina

    TPin* CfgInPin;               // vstupný konfiguračný pin
    TPin* CfgOutPin;              // výstupný konfiguračný pin
    TPin* InPin;                  // vstupný pin
    TPin* OutPin;                 // výstupný pin
}
```

```

bool StopFlag;                // indikátor zastavenia

int HorizontalReduction();     // pomocné funkcie...
int VerticalReduction();
int HVReduction();
int HorizontalExpansion();
int VerticalExpansion();
int HVExpansion();

int setConfigFromPin(TPin *pin); // konfigurácia na pine...
int putConfigOnPin(TPin *pin);
int freeConfigOnPin(TPin *pin);
int clearOutputPins();          // čistenie výstupných pinov...
int freeOutputData();

public:
    TReductionFilter();          // konštruktor
    TReductionFilter(HINSTANCE hInst); // konštruktor s parametrami
    ~TReductionFilter();         // deštruktor

int setConfigData(TBuffer config,int type); // predefinované funkcie triedy TFilter...
TBuffer getConfigData(int type);
int initialize();
int run();
int reset();
int stop();
int finalize();
int showConfigDialog();
};

```

7. Príklad použitia

