

Resample

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 *Resample* prevzorkuje vstupný šedotónový obraz na cieľové rozlíšenie alebo s použitím nastavených faktorov. Na prevzorkovanie je použitý nastavený konvolučný filter. Vstupom aj výstupom je šedo tónový obraz v štruktúre *TPlane*.

2. Závislosti

Trieda *Resample* používa tieto externé definičné súbory a knižnice:

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

Libs: *filtergraph.lib*

3. Implementačné informácie

Informácie o triede:

Názov: *Resample*

Verzia: *1.0*

Magic: *120*

Informácie o definíciách:

Header: *resample.h*

Konfiguračný header: *resamplecfg.h*

Lib: *resample.lib*

Informácie o knižnici obsahujúcej triedu:

Názov knižnice: *Resample Library*

Verzia knižnice: *1.0*

Dll súbor knižnice: *resample.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 prevzorkovaná šedo tónová rovina v štruktúre *TPlane*.

5. Konfigurácia

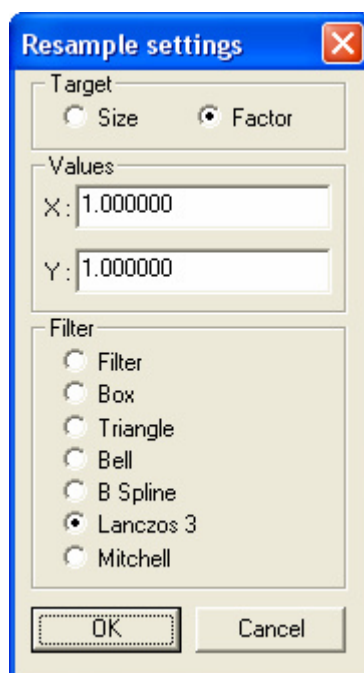
5.1 Konfiguračná štruktúra

Konfiguračná štruktúra je definovaná nasledovne:

```
typedef struct {  
    int      Mode;           // mód rozlíšenia  
    double   XVal;           // x-ová hodnota, buď rozlíšenie alebo faktor  
    double   YVal;           // y-ová hodnota, buď rozlíšenie alebo faktor  
    int      Filter;         // konvolučný filter  
} TResampleConfig;  
  
// módy rozlíšenia  
RM_TARGETSIZE      0      // hodnoty XVal a YVal predstavujú cieľové rozlíšenie  
RM_TARGETFACTOR    1      // hodnoty XVal a YVal predstavujú faktory zmeny  
  
// konvolučné filtre  
RF_FILTER          0      // filter  
RF_BOX             1      // box filter, obdĺžnikový filter  
RF_TRIANGLE        2      // trojuholníkový filter  
RF_BELL            3      // bell filter, zvonovitý filter  
RF_BSPLINE         4      // B-Spline filter  
RF_LANCZOS3        5      // filter Lanczos 3  
RF_MITCHELL        6      // Mitchellov filter
```

Hodnota *Mode* určuje, čo znamenajú hodnoty *XVal* a *YVal*. V móde *RM_TARGETSIZE* hodnoty *XVal* a *YVal* predstavujú cieľové rozlíšenie (zaokrúhlené na celé čísla). V móde *RM_TARGETFACTOR* predstavujú koeficienty zmeny rozlíšenia v daných smeroch. Pre hodnoty z intervalu $<0.0, 1.0)$ ide o zmenšenie rozlíšenia a pre hodnoty väčšie ako 1.0 ide o zväčšenie rozlíšenia. Na prevzorkovanie je použitý zvolený konvolučný filter.

5.2 Konfiguračný dialóg



Konfiguračný dialóg poskytuje voľby popísané pri konfiguračnej štruktúre. V bloku *Target* si zvolíme či hodnoty *X* a *Y* predstavujú cieľové rozlíšenie (voľba *Size*) alebo faktory (voľba *Factor*). V bloku *Values* nastavíme hodnoty *X* (*XVal*) a *Y* (*YVal*). V bloku *Filter* nastavíme konvolučný filter.

6. Definícia triedy

Trieda *Resample* je definovaná nasledovne:

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

    int        Mode;               // konfigurácia...
    double     XVal;
    double     YVal;
    int        Filter;

    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 resample(FilterFunc filterf, double fwidth); // funkcia prevzorkovania

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

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

