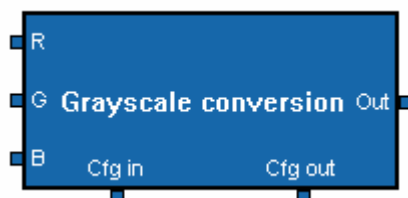


Grayscale conversion

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 *Grayscale conversion* prevedie vstupný obraz vo farebnom modeli RGB na šedo tónovú škálu. Vstupom sú tri šedo tónové obrazy v štruktúrach *TPlane*, v rovnakom rozlíšení, predstavujúce farebné roviny R, G a B. Výstupom je šedo tónový obraz v štruktúre *TPlane*, získaný z týchto troch rovín pomocou nastavenej metódy.

2. Závislosti

Trieda *Grayscale conversion* používa tieto externé definičné súbory a knižnice:

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

Libs: *filtergraph.lib*

3. Implementačné informácie

Informácie o triede:

Názov: *Grayscale conversion*

Verzia: *1.0*

Magic: *90*

Informácie o definíciách:

Header: *grayscaleconv.h*

Konfiguračný header: *grayscaleconvcfg.h*

Lib: *grayscaleconv.lib*

Informácie o knižnici obsahujúcej triedu:

Názov knižnice: *Grayscale conversion Library*

Verzia knižnice: *1.0*

Dll súbor knižnice: *grayscaleconv.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] R: Červený vstup. Vstupom je šedo tónová rovina v štruktúre *TPlane*.

[3] G: Zelený vstup. Vstupom je šedo tónová rovina v štruktúre *TPlane*.

[4] B: Modrý vstup. Vstupom je šedo tónová rovina v štruktúre *TPlane*.

[3] Out: Výstup. Výstupom je šedo tónová rovina v štruktúre *TPlane* získaná prevodom rovín R, G a B.

5. Konfigurácia

5.1 Konfiguračná štruktúra

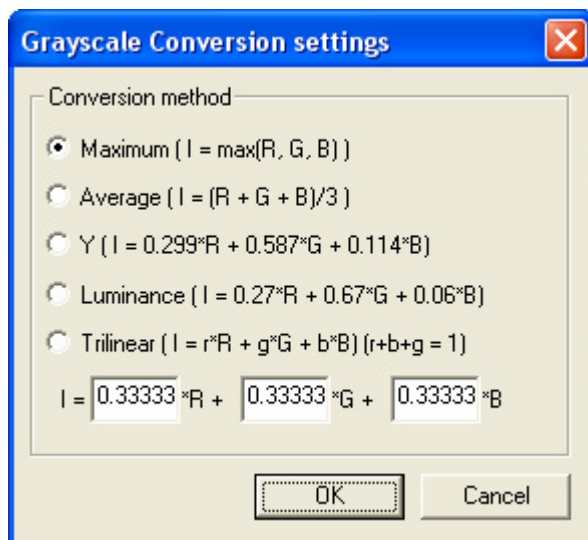
Konfiguračná štruktúra je definovaná nasledovne:

```
typedef struct {
    int          Conversion;           // metóda konverzie
    DET_FLOAT    Rc;                   // koeficient Rc pre trilineárnu konverziu
    DET_FLOAT    Gc;                   // koeficient Gc pre trilineárnu konverziu
    DET_FLOAT    Bc;                   // koeficient Bc pre trilineárnu konverziu
} TGrayscaleConversionConfig;

// Metódy konverzie
CONV_MAX      0    // I = max3(R, G, B) , maximálna hodnota z farebných zložiek
CONV_AVERAGE  1    // I = (R+G+B)/3 , priemerná hodnota farebných zložiek
CONV_Y         2    // I = Y = 0.289*R + 0.587*G + 0.114*B , kanál Y v móde YUV
CONV_LUM       3    // I = Lum = 0.27*R + 0.67*G + 0.06*B , luminancia
CONV_TRILINEAR 4    // I = Rc*R + Gc*G + Bc*B , kde Rc+Gc+Bc = 1 (nemusi),
// ľubovoľná trilineárna kombinácia farebných zložiek
```

Intenzita v maximovom móde (*CONV_MAX*) sa vypočíta ako maximum z farebných zložiek. Ostatné módy používajú trilineárnu metódu z preddefinovanými alebo nastaviteľnými koeficientmi. Mód priemerovania (*CONV_AVERAGE*) počíta intenzitu ako priemernú hodnotu z farebných zložiek (trilineárny mód pre koeficienty $Rc=Gc=Bc=1/3$). Mód Y (*CONV_Y*) počíta intenzitu ako hodnotu Y použitú vo farebných modeloch YUV, YIQ alebo YCbCr (trilineárny mód pre koeficienty $Rc=0.289$, $Gc=0.587$, $Bc=0.114$). Mód luminancie (*CONV_LUM*) počíta intenzitu ako hodnotu luminancie používanú pri rôznych algoritmoch (trilineárny mód pre koeficienty $Rc=0.27$, $Gc=0.67$, $Bc=0.06$). Trilineárny mód počíta intenzitu ako trilineárnu kombináciu farebných zložiek s nastavenými koeficientmi Rc , Gc a Bc . Súčet týchto koeficientov by mal byť 1, ale nie je to podmienkou.

5.2 Konfiguračný dialóg



Konfiguračný dialóg poskytuje voľby popísané pri konfiguračnej štruktúre. V bloku *Conversion method* môžeme nastaviť metódu (mód) prevodu na šedo tónovú škálu. Na výber sú voľby *Maximum* (*CONV_MAX*), *Average* (*CONV_AVERAGE*), *Y* (*CONV_Y*), *Luminance* (*CONV_LUM*) a *Trilinear* (*CONV_TRILINEAR*). V spodných troch kolonkách môžeme nastaviť koeficienty pre trilineárny mód konverzie.

6. Definícia triedy

Trieda *Grayscale conversion* je definovaná nasledovne:

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

    int Conversion;                // mód konverzie
    DET_FLOAT Rc;                  // koeficient pre trilineárny mód
    DET_FLOAT Gc;                  // koeficient pre trilineárny mód
    DET_FLOAT Bc;                  // koeficient pre trilineárny mód

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

    TPlane** InPlanes;            // pole vstupných rovín
    TPlane OutPlane;              // výstupná rovina

    TPin* CfgInPin;               // vstupný konfiguračný pin
    TPin* CfgOutPin;              // výstupný konfiguračný pin
    TPin* RPin;                   // vstupný pin pre červenú rovinu
    TPin* GPin;                   // vstupný pin pre zelenú rovinu
    TPin* BPin;                   // vstupný pin pre modrú rovinu
    TPin* OutPin;                 // výstupný pin
```

```

bool StopFlag;                // indikátor zastavenia

int MaxConv();                // prevedie konverziu v maximovom móde
int TrilinearConv(TVec3 c);    // prevedie konverziu v trilineárnom móde

int freeOutPlane();           // uvoľní výstupnú rovinu

int setConfigFromPin(TPin *pin);    // nastaví konfiguráciu z pinu
int putConfigOnPin(TPin *pin);      // sprístupní konfiguráciu na pine
int freeConfigOnPin(TPin *pin);     // uvoľní konfiguráciu na pine
int clearOutputPins();              // vyčistí výstupné piny
int freeOutputData();               // uvoľní výstupné dáta

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

    int setConfigData(TBuffer config, int type); // nastaví konfiguráciu
    TBuffer getConfigData(int type);             // vráti konfiguráciu
    int initialize();                             // inicializácia
    int run();                                    // beh
    int reset();                                  // reset stavu
    int stop();                                   // ukončenie behu
    int finalize();                               // finalizácia
    int showConfigDialog();                       // vyvolá konfiguračný dialóg
};

```

7. Príklad použitia

