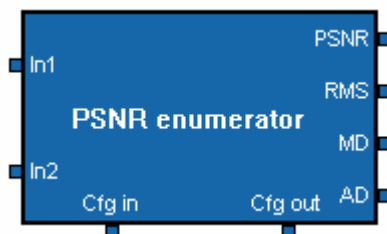


PSNR enumerator

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.....	2
6. Definícia triedy	3
7. Príklad použitia	4

1. Popis



Trieda *PSNR enumerator* vypočíta podobnosť dvoch šedo tónových obrazov na základe RMS metriky. Vstupom sú dva šedo tónové obrazy v štruktúrach *TPlane*. Výstupom sú štatistické hodnoty *PSNR* (*Peak Signal-to-Noise Ratio*), *RMS* (*Root Mean Square* – stredná kvadratická odchýlka), *MD* (*Maximal Difference*) – maximálna odchýlka obrazov meraná v rovnakom mieste, a *AD* (*Average Difference*) – priemerná odchýlka obrazov meraná v rovnakom mieste. Pri výpočte hodnoty *PSNR* sa za maximálnu hodnotu signálu považuje maximum dynamického rozsahu zo oboch vstupov. V prípade rôznosti dynamického rozsahu vstupov je možné nastaviť voľbu predškálovania (*Prescaling*), ktorá zabezpečí, že hodnoty vstupu *In2* sa budú škálovať do rozsahu vstupu *In1*.

2. Závislosti

Trieda *PSNR enumerator* používa tieto externé definičné súbory a knižnice:

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

Libs: *filtergraph.lib*

3. Implementačné informácie

Informácie o triede:

Názov: *PSNR enumerator*

Verzia: 1.0

Magic: 210

Informácie o definíciách:

Header: *psnrenumerator.h*

Konfiguračný header: *psnrenumeratorcfg.h*

Lib: *psnrenumerator.lib*

Informácie o knižnici obsahujúcej triedu:

Názov knižnice: *PSNR enumerator Library*

Verzia knižnice: *1.0*

Dll súbor knižnice: *psnrenumerator.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] In1: Prvý vstup. Vstupom je šedo tónová rovina v štruktúre *TPlane*.

[3] In2: Druhý vstup. Vstupom je šedo tónová rovina v štruktúre *TPlane*.

[4] PSNR: Výstup *PSNR*. Výstupom je hodnota *PSNR* vo formáte *DT_FLOAT*.

[5] RMS: Výstup *RMS*. Výstupom je hodnota *RMS* (*Root Mean Square*) vo formáte *DT_FLOAT*.

[6] MD: Výstup *Maximal Difference*. Výstupom je maximálna odchýlka obrazov meraná v rovnakom bode, vo formáte *DT_FLOAT*.

[7] AD: Výstup *Average Difference*. Výstupom je priemerná odchýlka obrazov meraná v rovnakom bode, vo formáte *DT_FLOAT*.

5. Konfigurácia

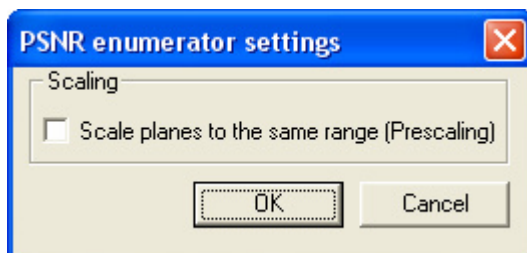
5.1 Konfiguračná štruktúra

Konfiguračná štruktúra je definovaná nasledovne:

```
typedef struct {  
    BOOL Prescaling;           // predškálovanie  
} TPSNREnumeratorConfig;
```

Voľba *Prescaling* povolí predškálovanie druhého vstupného obrazu na dynamický rozsah prvého obrazu.

5.2 Konfiguračný dialóg



Konfiguračný dialóg poskytuje voľby popísané pri konfiguračnej štruktúre. Voľba *Prescaling* povolí predškálovanie.

6. Definícia triedy

Trieda *PSNR enumerator* je definovaná nasledovne:

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

    TPlane* InPlane1;              // prvá vstupná rovina
    TPlane* InPlane2;              // druhá vstupná rovina
    DET_FLOAT OutPSNR;              // výstupná hodnota PSNR
    DET_FLOAT OutRMS;               // výstupná hodnota RMS
    DET_FLOAT OutMaxDif;           // výstupná hodnota Maximal Difference
    DET_FLOAT OutAverDif;          // výstupná hodnota Average Difference

    TPin* CfgInPin;                // vstupný konfiguračný pin
    TPin* CfgOutPin;               // výstupný konfiguračný pin
    TPin* In1Pin;                  // prvý vstupný pin
    TPin* In2Pin;                  // druhý vstupný pin
    TPin* OutPSNRPin;              // výstupné piny...
    TPin* OutRMSPin;
    TPin* OutMaxDifPin;
    TPin* OutAverDifPin;

    bool StopFlag;                 // indikátor zastavenia

    BOOL Prescaling;               // indikátor predškálovania

    int calc();                    // výpočet výstupných hodnôt

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

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

