

# Crop

## 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 *Crop* oreže okraje vstupného obrazu podľa nastavení a orezaný obraz dá na výstup. Vstupom je šedo tónový obraz v štruktúre *TPlane*. Výstupom je tiež šedo tónový obraz v štruktúre *TPlane*.

## 2. Závislosti

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

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

**Libs:** *filtergraph.lib*

## 3. Implementačné informácie

Informácie o triede:

**Názov:** *Crop*

**Verzia:** *1.0*

**Magic:** *110*

Informácie o definíciách:

**Header:** *crop.h*

**Konfiguračný header:** *cropcfg.h*

**Lib:** *crop.lib*

Informácie o knižnici obsahujúcej triedu:

**Názov knižnice:** *Crop Library*

**Verzia knižnice:** *1.0*

Dll súbor knižnice: *crop.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 orezaná šedo tónová rovina v štruktúre *TPlane*.

## 5. Konfigurácia

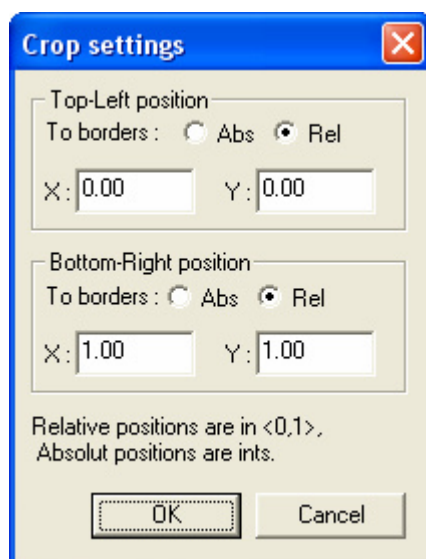
### 5.1 Konfiguračná štruktúra

Konfiguračná štruktúra je definovaná nasledovne:

```
typedef struct {  
    int    TLMode;      // pozičný mód ľavého horného rohu orezávacieho okna  
    int    BRMode;      // pozičný mód pravého dolného rohu orezávacieho okna  
    double Posx1;       // ľavá súradnica orezávacieho okna  
    double Posy1;       // horná súradnica orezávacieho okna  
    double Posx2;       // pravá súradnica orezávacieho okna  
    double Posy2;       // dolná súradnica orezávacieho okna  
} TCropConfig;  
  
//Pozíčné módy  
PM_ABSTOBORDER    0    // absolútny mód  
PM_RELTOBORDER    1    // relatívny mód
```

V absolútnom pozičnom móde (*PM\_ABSTOBORDER*) sú hodnoty *Posx1*, *Posy1*, alebo *Posx2*, *Posy2* brané ako súradnice orezávacieho okna (zaokrúhlené na celé číslo). V relatívnom pozičnom móde (*PM\_RELTOBORDER*) sú súradnice brané ako koeficienty, ktorými sa násobí rozlíšenie vstupu (t.j. ak ľavý horný roh je v relatívnom móde, tak jeho pozícia vo vstupnom obraze bude  $Posx1 * Cx$ ,  $Posy1 * Cy$  kde *Cx* a *Cy* sú rozmery vstupného obrazu). Pozičný mód je možné nastaviť osobitne pre ľavý horný roh okna a pravý dolný roh okna. V prípade, že niektorá zo súradníc orezávacieho okna presahuje rozlíšenie vstupného obrazu, beh je ukončený a vrátený je chybový kód.

## 5.2 Konfiguračný dialóg



Konfiguračný dialóg poskytuje voľby popísané pri konfiguračnej štruktúre. V bloku *Top-Left position* môžeme nastaviť pozíciu ľavého horného rohu orezávacieho okna. Hodnoty *X* a *Y* znamenajú x-ovú a y-ovú súradnicu. Voľbou *To borders* môžeme nastaviť pozičný mód rohu. Voľba *Abs* predstavuje absolútny mód (*PM\_ABSTOBORDER*) a voľba *Rel* predstavuje relatívny mód (*PM\_RELTOBORDER*). V bloku *Bottom-Right* môžeme rovnakým spôsobom nastaviť parametre pravého dolného rohu orezávacieho okna. Voľby potvrdíme tlačidlom *OK* alebo stornujeme tlačidlom *Cancel*.

## 6. Definícia triedy

Trieda *Crop* je definovaná nasledovne:

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

    int  TLMode;                   // pozičný mód ľavého horného rohu
    int  BRMode;                   // pozičný mód pravého dolného rohu
    double Posx1;                  // ľavá súradnica orezávacieho okna
    double Posy1;                  // horná súradnica orezávacieho okna
    double Posx2;                  // pravá súradnica orezávacieho okna
    double Posy2;                  // dolná súradnica orezávacieho okna

    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 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:
    TCropFilter();                          // konštruktor
    TCropFilter(HINSTANCE hInst);            // konštruktor s identifikátorom knižnice
    ~TCropFilter();                          // 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();                                // zastavenie behu
int finalize();                             // finalizácia
int showConfigDialog();                     // vyvolá konfiguračný dialóg
};

```

## 7. Príklad použitia

