

# Techniky mapování tónů

## 1 Popis tříd

Pro vývoj technik mapování tónů jsou implementovány následující dvě třídy. První třída zajišťuje práci se soubory ve formátu TIFF, které obsahují obrázky s širokým rozsahem jasů. Druhá třída slouží jako výchozí rodičovský objekt budoucím operátorům.

### 1.1 Třída **TMOImage**

Konstruktory a destruktory :

**TMOImage** ( ) Jednoduchá varianta konstruktoru. Inicializuje vnitřní proměnné – nastaví nulové rozměry obrázku, rozlišení a kalibrace nastaví na běžně používané hodnoty. Název souboru je přednastaven na *output.tif*.

**TMOImage** (const char \*filename) slouží k otevření zadaného souboru. Název souboru je předán v parametru filename. Jde o konstruktor, který po inicializaci vnitřních proměnných zavolá metodu *Open*.

**~TMOImage** ( ) uvolní alokovanou paměť a uzavře soubor.

Metody :

**Open** (const char \*filename)

Program se pokusí otevřít zadaný soubor ve standardním umístění. Jestliže nenajde soubor se shodným jménem, pokusí se otevřít soubory s příponou *.tif* nebo *.tiff*. Je možné načítat obrázky ve formátu tif a obrázky ze systému Radiance. Data ze souboru načte a vhodně překóduje do vnitřní reprezentace. Při neúspěšném otevření souboru se vyvolá výjimka číslo -1. Obrázek, který nemá uloženou informaci o použitém kódování vyvolá výjimku č. -2. Při načítání obrázku, který není uložen v LogLUV kódování se vyvolá výjimka č. -3. Vadný formát obrázku indikuje výjimka č. -4. Obrázek, který nemá zadané rozměry vyvolá výjimku č. -5. Při selhání načítání dat se vyvolá výjimka č. -8.

**New** (int width, int height, int format)

Metoda slouží pro vytvoření nového obrázku dané velikosti. Šířka

obrázku je předána metodě parametrem *width*, výška obrázku parametrem *height*. Rozměry jsou zadány v pixelech. Funkce alokuje potřebnou paměť a nastaví vnitřní proměnné. Formát obrázku může být volen z těchto možností :

- TMO\_ RGB = 0  
Obrázek je uložen ve formátu po sobě jdoucích trojic čísel v dvojnásobné přesnosti (double). Každá trojice reprezentuje červenou, zelenou a modrou barevnou složku jednoho pixelu. Pixely jsou uloženy shora po řádcích.
- TMO\_ XYZ = 1  
Obrázek je opět uložen ve formátu po sobě jdoucích trojic čísel (double). Pixely jsou uloženy shora po řádcích.
- TMO\_ Yxy = 2  
Obrázek je uložen ve formátu po sobě jdoucích trojic čísel (double). Složka Y reprezentuje luminanci pixelu, x a y nese barevnou informaci.
- TMO\_ Y = 256  
Obrázek je uložen ve formátu po sobě jdoucích čísel (double). Slouží k uložení černobílých obrazů luminance.
- TMO\_ NOTSPEC  
Obrázek je uložen ve formátu blíže nespecifikovaných po sobě jdoucích trojic čísel (double).

Jméno souboru je předvoleno na *output.tif*.

**New** (const TMOImage &ref, int format)

Vytvoří nový obrázek na základě informací o jiném obrázku. Podle specifikovaného formátu překóduje referenční obrázek. Nový obrázek má stejné rozměry, obrazová data a jméno jako vzor *ref*.

**Close** ( )

Uzavře soubor, vynuluje vnitřní proměnné a nastaví počáteční hodnoty souboru.

**Save** ( )

Slouží k uložení souboru ve formátu Truecolor v barevné hloubce 24 bitů na pixel. Funkce uloží soubor pod stávajícím jménem. Jestliže nelze otevřít výstupní soubor vyvolá se výjimka č. -10. Jestliže při zápisu dojde k chybě, vyvolá se výjimka č. -11. Úspěšné uložení souboru je indikováno návratovou hodnotou 0.

**SaveAs** (const char \*filename)

Uloží obrázek pod novým jménem.

**SaveWithSuffix** (const char \*suffix)

Uloží obrázek pod výchozím názvem, ke kterému přidá zvolený řetězec *suffix*. Metoda je vhodná pro ukládání transformovaných obrázků.

**ProgressBar** (int part, int all)

Funkce se volá při načítání a ukládání souboru jako indikátor. *Part* reprezentuje poměrnou část celku – hotového procesu *all*.

**SetProgress** (int( \*)(TMOImage \*, int, int))

Nastaví callback funkci indikátoru.

**WriteLine** (const wchar\_t \* text)

Slouží jako textový výstup operátoru. *Text* je ve formátu řetězce Unicode.

**SetWriteLine** (int( \*)(TMOImage \*, const wchar\_t \*))

Nastaví callback funkci textového výstupu.

**GetData** ( )

Funkce vrací ukazatel na datovou oblast obsahující informace o jednotlivých pixelech obrázku.

**GetWidth** ( )

Vrací šířku obrázku v pixelech.

**GetHeight** ( )

Vrací výšku obrázku v pixelech.

**GetOffset** (int offset)

Funkce vrací ukazatel na datovou oblast začínající pixelem v pořadí daném číslem *offset*.

**GetPixel** (int x, int y)

Funkce vrací ukazatel na datovou oblast začínající pixelem na pozici dané souřadnicemi  $[x, y]$ .

**GetStonits** ( )

Vrací skutečnou hodnotu luminance mapovanou na hodnotu 1.

**Convert** (int format)

Funkce převede obrázek do formátu specifikovaném parametrem *format*. Parametr může nabývat jednu z hodnot popsaných ve funkci *New*.

## 1.2 Třída TMO

Slouží jako výchozí třída pro zděděné objekty jednotlivých operátorů. Obsahuje abstraktní virtuální metodu *Transform*, která zajišťuje vlastní transformaci.

Konstruktory a destruktory :

**TMO** ( ) je základní konstruktor, provádí vynulování počátečních hodnot.

**TMO** (int x, int y, char \*source, char \*dest) je konstruktor předávající nezbytné údaje o obrázku. Parametry *x* a *y* jsou rozměry obrázku, ukazatele *source* a *dest* směřují na vstupní resp. transformovaná data.

**~CTMO** ( ) uvolňuje paměť.

Metody :

**SetSource** (TMOImage& p) nastavuje data vstupního obrázku.

**GetSource** (TMOImage \* \* p) vrací ukazatel na vstupní obrázek.

**SetDestination** (TMOImage& p) nastavuje data transformovaného obrázku.

**GetDestination** (TMOImage \* \* p) vrací ukazatel na výstupní obrázek.

**SetImage** (TMOImage& p) nastavuje data vstupního a zároveň výstupního obrázku.

**GetImage** (TMOImage \* \* p) vrací ukazatel na zpracovávaný obrázek.

**SetImages** (TMOImage&, TMOImage&) nastavuje data vstupního a výstupního obrázku.

**GetImages** (TMOImage \* \*, TMOImage \* \*) vrací ukazatele na vstupní a výstupní obrázek.

**SetDimensions** (int x, int y) nastavuje rozměry obrázku v pixelech.

**GetDimensions** (int \*x, int \*y) vrací rozměry obrázku v pixelech.

**GetMaxAvg** (double \*maximum, double \*average) vrací maximální a průměrnou hodnotu jasu vstupního obrázku.

**SetLuminance** (int x, int y, double L) nastavuje luminanci bodu na pozici  $[x, y]$  na hodnotu  $L$ .

**GetLuminance** (int x, int y, int r = 0) vrací luminanci v okolí bodu s poloměrem  $r$  na pozici  $[x, y]$ .

**SetName** (const wchar\_t \* name) nastavuje jméno operátoru.

**GetName** () vrací jméno operátoru.

**SetDescription** (const wchar\_t \* name) nastavuje popis metody.

**GetDescription** () vrací popis metody.

**Register** (TMOPParameter& p) zaregistruje zadaný parametr – označí danou proměnnou jako vstupní parametr mapování.

Parametry :

**TMOImage \* pSrc** ukazatel na vstupní obrázek.

**TMOImage \* pDst** ukazatel na výstupní obrázek.

**int iWidth** šířka obrázku v pixelech.

**int iHeight** výška obrázku v pixelech.