Tech document for Image Editing

This file includes the detail description of functions that I used in the TargaImage.cpp from project Image Editing.

Constructors

TargaImage ::TargaImage(void)

Default constructor. Create an empty image.

TargaImage :: TargaImage(int w, int h)

Create an empty image with given width and height.

Parameters

- **w** width of the image
- **h** height of the image

TargaImage::TargaImage(int w, int h, unsigned char *d)

Create an image with given width, height, and content of image.

Parameters

- **w** width of the image
- **h** height of the image
- **d** content of image, is consist of RGBA values.

TargaImage::TargaImage(const TargaImage& image)

Copy constructor. Copy from another given image.

Parameters

- **image** image to copy

Destructor

TargaImage::~TargaImage()

Destructor. Free memory.

Methods

TargaImage* TargaImage::Load_Image(char *filename)

Load a targa image to edit.

Parameters

- **filename** name of the image to be loaded

Return

Return a new TargaImage object which must be deleted by caller. Return NULL on failure.

bool TargaImage::Save_Image(const char* filename)

Save image to a file

Parameters

filename name of the image to be saved

Return

true if saving is successful

unsigned char* TargaImage::To_RGB(void)

Converts an image from RGBA to RGB form.

Return

rgb pixel data - 24 bits per pixel.

Also see

TargaImage::RGBA_To_RGB()

void TargaImage::RGBA_To_RGB(unsigned char *rgba, unsigned char *rgb)

Given a single RGBA pixel return, via the second argument, the RGB equivalent composited with a black background.

TargaImage* TargaImage::Reverse_Rows(void)

Copy this into a new image, reversing the rows as it goes.

Return

A pointer to the new image is returned.

void TargaImage::ClearToBlack()

Clear the image to all black.

bool TargaImage::To_Grayscale()

Convert image to grayscale. Red, green, and blue channels should all contain grayscale value. Alpha channel should be left unchanged.

Return

True if gray scaling is successful

bool TargaImage::Quant_Uniform()

Convert the image to an 8-bit image using uniform quantization.

Return

True if uniform quantization is successful bool TargaImage::Dither_Threshold()

Dither the image using a threshold of 1/2

Return

True if dithering is successful

bool TargaImage::Dither_Random()

Dither image using random dithering.

Return

True if dithering is successful

bool TargaImage::Dither_Bright()

Dither the image while conserving the average brightness.

Return

True if dithering is successful

bool TargaImage::Dither_Cluster()

Perform clustered differing of the image.

Return

True if dithering is successful

void TargaImage::filter(float* filter_matrix, float divide, int kernel_size)

Main body of all filter function.

Parameters

filter_matrix the mask to filter
divide the divisor to divide the weighted sum after filtering
kernel_size the size of the mask

float TargaImage::filter_pixel(float* filter_matrix, float divide, int kernel_width, int kernel_height, int w, int h, int channel)

helper function to generate value of each pixel for filter

Parameters

filter_matrix the mask to filter
divide the divisor to divide the weighted sum after filtering
kernel_width the width of the mask
kernel_height the height of the mask
w the x position of the kernel in data
h the y position of the kernel in data
channel the rgba channel

float TargaImage::findKernelValue(float* kernel, int x, int y, int kernel_width, int kernel_height)

helper function to fill kernel matrix for filter

Parameters

- **kernel** the incomplete kernel to be fill in
- **x** current x position in the kernel
- **y** current y position in the kernel
- **kernel_width** the width of the kernel
- **kernel_height** the height of the kernel

bool TargaImage::Filter_Box()

Perform 5x5 box filter on this image

Return

True if filtering is successful

bool TargaImage::Filter_Bartlett()

Perform 5x5 Bartlett filter on this image.

Return

True if filtering is successful

bool TargaImage::Filter_Gaussian()

Perform 5x5 Gaussian filter on this image

Return

True if filtering is successful

bool TargaImage::Half_Size()

Halve the dimensions of this image

Return

True if succeed

bool TargaImage::Double_Size()

Double the dimensions of this image.

Return

True if succeed

bool TargaImage::Rotate(float angleDegrees)

Rotate the image clockwise by the given angle.

Return

True if succeed