OpenCV Tutorial C++

Filtering Images

Image filtering is an important part of computer vision. For most of computer vision applications, filtering should be done before anything else. OpenCV supports lots of in-build filtering methods for images. Here is the list of filtering methods that I am going to discuss with you in the following posts (with OpenCV 2.4.5 and C++)

- **Change Brightness of Image or Video**
- **Change Contrast of Image or Video**
- **Histogram Equalization of Grayscale or Color Image**
- **Smooth / Blur Images**

Here is the list of image filtering methods which are explained using examples with OpenCV 2.1 in C style (not C++)

- Eroding
- Dilating
- Inverting

Here is the original image which I am going to filter using above methods.

Original Image

If you have not install and configure OpenCV yet, please refer to Installing & Configuring with Visual Studio.

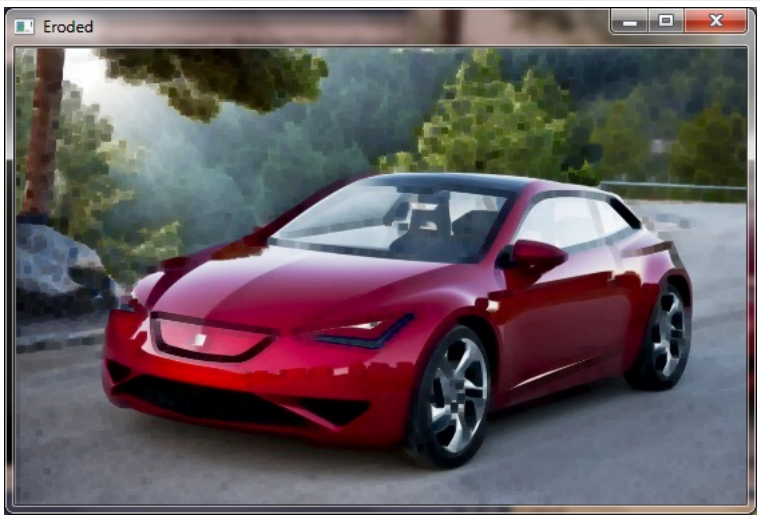
Eroding

Eroding is a simple way of filtering images. Here is how it can be done with OpenCV.

//

```
#include "stdafx.h"
#include <cv.h>
#include <highgui.h>

int main()
{
    //display the original image
    IplImage* img = cvLoadImage("C:/MyPic.jpg");
    cvNamedWindow("MyWindow");
    cvShowImage("MyWindow", img);

    //erode and display the eroded image
    cvErode(img, img, 0, 2);
    cvNamedWindow("Eroded");
    cvShowImage("Eroded", img);

    cvWaitKey(0);

    //cleaning up
    cvDestroyWindow("MyWindow");
    cvDestroyWindow("Eroded");
    cvReleaseImage(&img);

    return 0;
}
//////////////////////////////////////////////////////////////////////////////////
```

You can download this OpenCV visual c++ project from here.



Eroded Image

New OpenCV functions which are not found earlier are explained here

- **cvErode(img, img, 0, 2)**

The 1st parameter is the source image.
The 2nd parameter is the destination image which is to be the eroded image.
Here the 3rd parameter is the structuring element used for erosion. If it is `0`, a 3×3 rectangular structuring element is used.
The 4th parameter is the number of times, erosion is applied.
This function can process images in place. That means same variable can be used for the 1st and 2nd parameters.

If you want more explanation about various methods in the above computer application , please refer to Capturing Images & Videos.

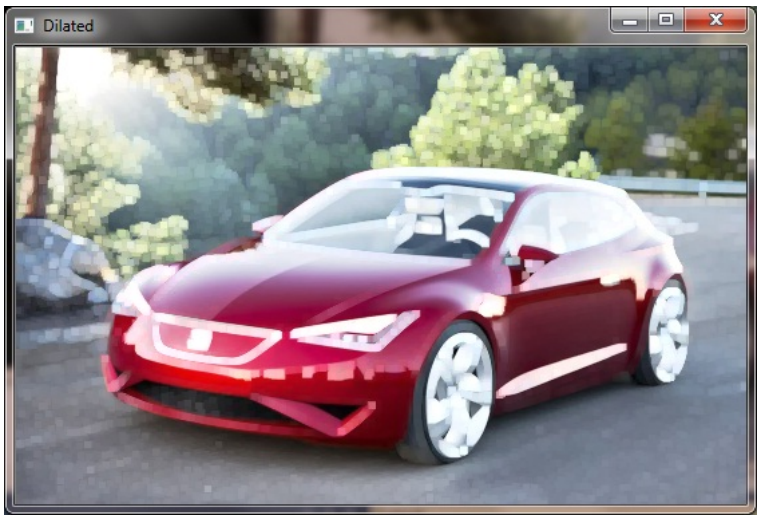
Dilating

Dilating is something like opposite of the eroding an image. Here is the OpenCV code.

```
//////////////////////////////////////////////////////////////////////////////////
#include "stdafx.h"
#include <cv.h>
#include <highgui.h>

int main()
{
    //display the original image
    IplImage* img = cvLoadImage("C:/MyPic.jpg");
    cvNamedWindow("MyWindow");
    cvShowImage("MyWindow", img);
```

```
	//dilate and display the dilated image
	cvDilate(img, img, 0, 2);
	cvNamedWindow("Dilated");
	cvShowImage("Dilated", img);

	cvWaitKey(0);

	//cleaning up
	cvDestroyWindow("MyWindow");
	cvDestroyWindow("Dilated");
	cvReleaseImage(&img);

	return 0;
}
//////////////////////////////////////////////////////////////////////////////
```

You can download this OpenCV visual c++ project from here.



Dilated Image

New OpenCV functions which are not found earlier are explained here

- **cvDilate(img, img, 0, 2)**

The 1st parameter is the source image.
The 2nd parameter is the destination image which is to be the dilated image.
Here the 3rd parameter is the structuring element used for dilation. If it is `0`, a 3×3 rectangular structuring element is used.
The 4th parameter is the number of times, dilation is applied.
This function can process images in place. That means same variable can be used for the 1st and 2nd parameters.

Inverting

Inverting an image is like taking the negative of an image.

```
//////////////////////////////////////////////////////////////////////////////


#include "stdafx.h"
```

```
#include <cv.h>
#include <highgui.h>

int main()
{
	//display the original image
	IplImage* img = cvLoadImage("C:/MyPic.jpg");
	cvNamedWindow("MyWindow");
	cvShowImage("MyWindow", img);

	//invert and display the inverted image
	cvNot(img, img);
	cvNamedWindow("Inverted");
	cvShowImage("Inverted", img);

	cvWaitKey(0);

	//cleaning up
	cvDestroyWindow("MyWindow");
	cvDestroyWindow("Inverted");
	cvReleaseImage(&img);

	return 0;
}
```

//

You can download this OpenCV visual c++ project from here.

Inverted Image

New OpenCV functions which are not found earlier are explained here

- **cvNot(img, img)**

This function inverts every bit in every element of the image in the 1st parameter and places the result in the image in the 2nd parameter. This function can process images in place. That means same variable can be used for the 1st and 2nd parameters.

e.g - For a 8 bit image, the value 0 will be mapped to (255-0)=255
the value 46 will be mapped to (255-46)=209

For a 16 bit image, the value 0 will be mapped to (65535-0)=65535
the value 46 will be mapped to (65535-46)=65489

Next Tutorial : How to Add Trackbar

Previous Tutorial : Write Image & Video to File

Posted by Shermal Fernando