

[Forums / Grupos de estudio](#)[Help Center](#)

## Grupo C++

[Subscribe for email updates.](#)No tags yet. [+ Add Tag](#)Sort replies by: [Oldest first](#) [Newest first](#) [Most popular](#)[Angel Vivanco](#) · 2 months ago [🔗](#)

Para todos los interesados en trabajar con C++

[↑ 1](#) [↓](#) · [flag](#)[Milton Wong](#) · 2 months ago [🔗](#)

OpenCV C++, the right one?

[↑ 3](#) [↓](#) · [flag](#)[+ Comment](#)[Fernando Domínguez Pousa](#) Signature Track · 2 months ago [🔗](#)

Me uno al hilo. He trabajado con MATLAB, y tengo curiosidad por C++ y OpenCV.

[↑ 0](#) [↓](#) · [flag](#)[+ Comment](#)[Milton Wong](#) · 2 months ago [🔗](#)

Good, maybe we can wait for the example code from this class, and then "translate" it into OpenCV C++

[↑ 1](#) [↓](#) · [flag](#)[+ Comment](#)[Angel Vivanco](#) · 2 months ago [🔗](#)

Si la idea es realizar los ejercicios y ejemplos del curso en C/C++ usando OpenCV, si alguien ha usado otras librerías sería interesante que nos cuente su experiencia.

El tema de los entornos de desarrollo tiene su propio foro y existen muchas variantes pero si alguien aun no lo tiene preparado tambien podriamos ver como ayudar.

Saludos

↑ 0 ↓ · flag

[+ Comment](#)



Fernando Domínguez Pousa Signature Track · 2 months ago [🔗](#)

A mi me gustaría trabajar con el IDE de Eclipse. Me parece un entorno bastante versátil y cómodo. ¿Sabéis si conlleva mucha dificultad trabajar con OpenCV y Eclipse?

↑ 0 ↓ · flag

[+ Comment](#)



Milton Wong · 2 months ago [🔗](#)

. For the reason of supporting more OS (Either Linux, MacOS, Windows), I choose rather than eclipse + opencv others like visual studio.

I Had Followed the installation guide by the following, and it works :)

or windows: <https://www.youtube.com/watch?v=kVyVEHK4qfQ>

For ubuntu <https://www.youtube.com/watch?v=0jYCbK7vn4k>

↑ 2 ↓ · flag



Fernando Domínguez Pousa Signature Track · 2 months ago [🔗](#)

Thanks! I tried with first for Windows and it worked without problem. A little bit confused with cmd, however it worked! ;)

↑ 0 ↓ · flag

[+ Comment](#)

Ione Monroy · 2 months ago [🔗](#)

Yo trabajo con Code:Blocks, compilando las librerías de OpenCV y jamás he tenido ningún problema, os lo recomiendo.

↑ 1 ↓ · flag

+ Comment

David Vázquez Bermúdez STAFF · 2 months ago

Buenas,

si queréis probar los conceptos del curso en C++ definitivamente os recomiendo usar OpenCV. Ahí encontraréis practicamente todos los detecteres que se explicarán en el curso ya programados y entrenados. Podréis encontrar:

- Viola Jones: AdaBoost Haar classifier
- HOG SVM: Histogram of oriented gradients
- Latent SVM: Deformable part based model
- LBP

Además el adaboost y el HOG incluso lo encontraréis implementado en CUDA para que funcionen en el hardware de NVIDIA.

Hay varios modelos entrenados como para caras, personas, tazas...

Suerte

↑ 2 ↓ · flag

+ Comment

Luis Gabriel Jaimes · 2 months ago

Manana tratare de integrar OpenCV con Eclipse EDT, excelente idea trabajar in C++.

↑ 1 ↓ · flag

+ Comment

Ernesto de la Cruz Guevara Ramírez · 2 months ago

Ok me uno también a este grupo y espero la pasemos bien, yo he trabajado con OpenCV en C++, Visual Studio y con QtCreator en Windows (MingW) y Linux (GNUC).

↑ 1 ↓ · flag

+ Comment

Luis Gabriel Jaimes · 2 months ago

Ayer, siguiendo al pie de la letra este enlace, pude integrar Eclipse y openCV 2.4.10:

<http://thinkingweb.co/configure-mingw-eclipse-ide-for-cc-developers-and-opencv-2-4-10/>

↑ 1 ↓ · flag

---

[+ Comment](#)

Luis Gabriel Jaimes · 2 months ago 

Olvide decir que mi sistema operativo es windows 7, así que instale eclipse luna cdt, openCV 2.4.10 in windows 7 y hasta ahora luce muy bien: <http://thinkingweb.co/configure-mingw-eclipse-ide-for-cc-developers-and-opencv-2-4-10/>

↑ 0 ↓ · flag

---

[+ Comment](#)

Anonymous · 2 months ago 

Buenas compañeros, si alguien me puede ayudar les explico mi problema. Haciendo el ejercicio del video de la piramide con ventana deslizable no obtengo los mismos resultados, sospecho que es causado por el truncamiento pero no logro encontrar la solución.

En el video se obtienen estos valores:

Nivel 0 3285 ventanas  
Nivel 1 2065 ventanas  
Nivel 2 1248 ventanas  
Nivel 3 741 ventanas  
Nivel 4 403 ventanas  
Nivel 5 225 ventanas  
Nivel 6 95 ventanas  
Nivel 7 15 ventanas

Total = 8077

y en mi programa sale

```
Nivel 0 image width = 640 image height = 480
Ventanas 3285
Nivel 1 image width = 533 image height = 400
Ventanas 2065
Nivel 2 image width = 444 image height = 333
Ventanas 1248
Nivel 3 image width = 370 image height = 277
Ventanas 741
Nivel 4 image width = 308 image height = 230
Ventanas 403
Nivel 5 image width = 256 image height = 191
Ventanas 200
Nivel 6 image width = 213 image height = 159
Ventanas 76
Nivel 7 image width = 177 image height = 132
Ventanas 15
Total ventanas = 8033
Process returned 0 (0x0) execution time : 0.172 s
Press any key to continue.
```

y mi codigo es

```
int getSlidingWindows(int imgWidth, int imgHeight, int winCanWidth, int winCanHeight, int dx, int dy, double s)
{
    int sumWindows = 0;
    int windowsInX = 0;
    int windowsInY = 0;
    int windowsPerLevel = 0;
    int level = 0;

    while(true) {
        cout << "Nivel " << level << " image width = " << imgWidth << " image height = " << imgHeight << endl;

        windowsInX = ((imgWidth - winCanWidth)/dx) + 1;
        windowsInY = ((imgHeight - winCanHeight)/dy) + 1;

        windowsPerLevel = windowsInX * windowsInY;

        cout << "Ventanas " << windowsPerLevel << endl;

        sumWindows += windowsPerLevel;

        // Escalando la imagen
        imgWidth = imgWidth / s;
        imgHeight = imgHeight / s;
        level++;

        if(imgWidth <= winCanWidth || imgHeight <= winCanHeight) {
            break;
        }
    }
}
```

```
    }  
  
    }  
  
    return sumWindows;  
}
```

Gracias y saludos

↑ 0 ↓ · flag

[+ Comment](#)

David Jiménez Fernández · 2 months ago 

Buenas,

Para obtener el "patrón" de las diapositivas lo más sencillo es no truncar (ignorar los decimales), sino redondear (si < 0.5, hacia abajo, si > 0.5, hacia arriba, en caso de justo 0.5 tener un criterio prefijado), y en cada paso dividir entre 1.2 pero no la versión redondeada (que es la que se usa), sino el valor con decimales acumulado. Me explico desde el principio:

$640 / 1.2 = 533.3333333$  --> Redondeo = 533 (porque la ventana tiene que tener un número entero de píxeles)

Si en cada paso volviéramos a dividir la versión entera, llegaríamos a un punto en el que redondear no nos daría el mismo resultado que en las transparencias. Veámoslo:

$533 / 1.2 = 444.166666$  --> Redondeo = 444

$444 / 1.2 = 370.0$  --> Redondeo = 370

$370 / 1.2 = 308.333333$  --> Redondeo = 308 -> Debería ser 309 según la transparencia. De ahí en adelante se acumula error.

Sin embargo si como resultado para calcular tamaños y números de ventana tomamos el redondeo, pero para seguir dividiendo aplicamos la versión con decimales....

$533.3333333 / 1.2 = 444.4444444$  --> Redondeo para otros cálculos = 444

$444.4444444 / 1.2 = 370.3703704$  --> Redondeo para otros cálculos = 370

$370.3703704 / 1.2 = 308.6419753$  --> Ahora el redondeo por ser > 308.5 es 309, como en la diapositiva.

Y lo mismo para todos los cálculos restantes.

Comentarios respecto al código:

a) Las líneas :

```
windowsInX = ((imgWidth - winCanWidth)/dx) + 1;  
windowsInY = ((imgHeight - winCanHeight)/dy) + 1;
```

quedan sujetas a errores de redondeo también, en cuanto la resta no sea múltiplo exacto de dx ó dy. En este caso concreto, como es contar cuántas ventanas "cabén", considero más apropiado tomar siempre el valor hacia abajo (función "floor") de la división y luego sumar 1.

```
windowsInX = floor((imgWidth - winCanWidth)/dx) + 1;  
windowsInY = floor((imgHeight - winCanHeight)/dy) + 1;
```

b) Redondear "bien" en C++ no es tan trivial como parece. Si el compilador no posee la función "round()" (C++11 o superior, creo), hay que usar librerías específicas de matemáticas porque hay sutilezas que pueden dar resultados inesperados si se trabaja con float, double, 32 bits, 64 bits, Windows, Linux... Una solución de andar por casa es hacer algo así ;

```
floor(x + 0.5);
```

pero no está recomendado del todo porque no funciona siempre igual (si hay números negativos, o justo en casos de .4999999 que el compilador haya truncado mal).

De todas formas estos comentarios son para reproducir los resultados de las transparencias, pero tampoco se ha insistido mucho en el cálculo exacto y no sabemos si ese ejemplo está así simplemente por cuestiones didácticas.

Un saludo.

↑ 1 ↓ · flag

[+ Comment](#)



Angel Vivanco · 2 months ago

Como comenta David la solución es algo dependiente de la plataforma (SO, compilador), por ahí hay un post que de momento no pude encontrar para poner el enlace, donde se discutía algo similar respecto a diferencias en resultados debidas a redondeos.

Tomando tu código y trabajando con gcc (en Windows7) hice este cambio y obtuve los mismos resultados del video.

```
// Escalando la imagen  
double tempWidth = imgWidth / s;  
double tempHeight = imgHeight / s;
```

```
imgWidth = round(tempWidth);  
imgHeight = round(tempHeight);
```

Si usas Microsoft Visual C++ tal ves necesites algo como esto

```
#include <cmath>  
inline double round(double x) { return (floor(x + 0.5)); }
```

Espero te sea de utilidad.

Saludos!

↑ 0 ↓ · flag

[+ Comment](#)

New post

To ensure a positive and productive discussion, please read our [forum posting policies](#) before posting.

<b>B</b>	<i>I</i>			Link	<code>&lt;code&gt;</code>	Pic	Math		Edit: Rich ▼	Preview
<div></div>										

☐ Make this post anonymous to other students

☒ Subscribe to this thread at the same time

Add post



