Práctica 1. Introducción al Manejo de Imágenes

M.I. Moisés García Villanueva

4 de marzo de 2014

1. Qué es Visión Computacional

Visión Computacional es una área muy amplia, su principal objetivo es realizar una transformación de datos, obtenidos a partir de una camara (fotografica o video), en una nueva representación de los datos o para tomar una decisión. Todas las transformaciones que se realicen son encaminadas para lograr una meta en particular.Una nueva transformación puede llevarnos a transformar una imagen de color a una imagen en escala de gris o remover el movimiento de la camara en una secuencia de imágenes.

Como regla general en el diseño de sistemas de visión computacional: mientras más restrictivo sea el contexto del sistema de visión computacional, más podremos realizar con estas restricciones para simplificar el problema y así lograremos tener una solución final lo más real posible.

1.1. Manejo de OpenCV

1. Programa para leer una imagen.

   a) Crear un archivo `practica01-1.c` con el siguiente contenido:

```
#include "highgui.h"

int main( int argc, char** argv ) {
        IplImage* img = cvLoadImage( argv[1] );
        cvNamedWindow( "Example1", CV_WINDOW_AUTOSIZE );
        cvShowImage( "Example1", img );
        cvWaitKey(0);
        cvReleaseImage( &img );
        cvDestroyWindow( "Example1" );
}
```

   b) Compilar de la siguiente forma:

```
g++ ‘pkg-config --cflags --libs opencv‘ practica01_1.c -o verimagen
```

   c) Ejecutar el programa con un argumento. El argumento del programa debe ser el nombre dearchivo de una imagen, por ejemplo ejecutar: `./verimagen carita.jpg`

2. Transformación Simple en una imagen.

   a) Crear un archivo con el nombre `practica01-2.c` con el siguiente contenido:

```
#include "cv.h"
#include "highgui.h"
void example2_4( IplImage* image ){
// Crear las ventanas para mostrar las im\’agenes de entrada y salida
//
   cvNamedWindow( "Example4-in" );
   cvNamedWindow( "Example4-out" );
// Crear unja ventana para mostrar la imagen de entrada
```

```
//
   cvShowImage( "Example4-in", image );
// Crear una nueva imagen en la que se colocara la imagen de salida suavizada
//
   IplImage* out = cvCreateImage( cvGetSize(image), IPL_DEPTH_8U, 3 );
// Ahora hacemos el suavizado de la imagen
//
   cvSmooth( image, out, CV_GAUSSIAN, 3, 3 );
// Mostrar la imagen suavizada en la ventana de salida
//
   cvShowImage( "Example4-out", out );
//
   cvReleaseImage( &out );
// Esperar a que el usuario presione una tecla, entonces eliminar las ventanas
//
   cvWaitKey( 0 );
   cvDestroyWindow( "Example4-in" );
   cvDestroyWindow( "Example4-out" );
}
int main( int argc, char** argv ) {
IplImage* img = cvLoadImage( argv[1] );
example2_4(img);
}
```

b) Compilar el programa:

```
g++ `pkg-config --cflags --libs opencv` practica01_2.c -o verimagen2
```

c) Ejecutar el programa con un argumento de entrada, el cual corresponde al nombre de archivo de una imagen, por ejemplo: `./verimagen2 imagen.jpg`

2. Manejo de colores RGB

Canales en una imagen RGB.

Referencias

[1] L. Enrique Sucar y Giovani Gómez, *Visión Computacional*.