

Reconocimiento de imágenes utilizando GPUs, aplicacion a fútbol de robots

Ignacio Eguinoa
Facultad de Informática, UNLP

3 de febrero de 2015

Resumen

En este trabajo se desarrollan conceptos relacionados con la vision por computadoras utilizando unidades de procesamiento gráfico(GPUs). Se incluye una descripción de la libreria OpenCV, incluyendo el módulo que implementa aceleración mediante GPUs. Además, se realiza una desarrollo práctico que consiste en reimplementar una libreria para el procesamiento de imágenes provenientes de fútbol de robots. Se plantean variaciones en esta libreria que aceleran distintos pasos del procesamiento de imagenes utlizando una GPU. Las distintas variantes son evaluadas utilizando un sistema de pruebas y los resultados analizados en base a las características de la arquitectura.

Índice general

1. Introduccion	2
1.1. Estructura del trabajo	2
1.2. La arquitectura GPU	2
2. Libreria OpenCV	3
3. Trabajo experimental	5
3.1. Futbol robot	5
3.2. Implementacion existente	5
3.2.1. La libreria bottracker	5
3.2.2. Programa cliente	5
3.3. Implementaciones sobre GPU	5
3.3.1. Resultados	5
4. Conclusiones y trabajo futuro	6

Capítulo 1

Introduccion

1.1. Estructura del trabajo

En este primer capitulo se realiza una introducción a la arquitectura GPU. El objetivo es dar una idea general de las características que posee y las posibilidades que ofrece tanto para procesamiento de gráficos como para cómputo de propósito general

En el capitulo 2 se describe de forma general la libreria OpenCV, principalmente los módulos y funciones que se utilizarán luego en el desarrollo.

El capitulo 3 contiene el desarrollo experimental del trabajo. En primer lugar se describe el contexto de la aplicación y las implementaciones existentes (libreria para procesamiento de imágenes de fútbol robot). Luego se plantean modificaciones sobre ésta, utilizando el módulo de GPU provisto por la libreria OpenCV. Se hacen evaluaciones de las distintas modificaciones y se analizan los resultados en base a los conceptos explicados en los capitulos previos.

1.2. La arquitectura GPU

Capítulo 2

Librería OpenCV

La librería OpenCV (Open Source Computer Vision Library: <http://opencv.org>) reúne una gran cantidad de algoritmos asociados a la visión por computadoras. Es una librería open-source que se distribuye bajo una licencia BSD. is an open-source BSD-licensed library that includes several hundreds of computer vision algorithms.

La librería está compuesta de distintos módulos que proveen funcionalidades independientes entre sí. Algunos módulos comunes son: `core` - a compact module defining basic data structures, including the dense multi-dimensional array `Mat` and basic functions used by all other modules. `imgproc` - an image processing module that includes linear and non-linear image filtering, geometrical image transformations (resize, affine and perspective warping, generic table-based remapping), color space conversion, histograms, and so on. `video` - a video analysis module that includes motion estimation, background subtraction, and object tracking algorithms. `calib3d` - basic multiple-view geometry algorithms, single and stereo camera calibration, object pose estimation, stereo correspondence algorithms, and elements of 3D reconstruction. `features2d` - salient feature detectors, descriptors, and descriptor matchers. `objdetect` - detection of objects and instances of the predefined classes (for example, faces, eyes, mugs, people, cars, and so on). `highgui` - an easy-to-use interface to video capturing, image and video codecs, as well as simple UI capabilities.

Módulo `core`:

OpenCV 2.0 introduced a new C++ interface which offered a new way of doing things which means you do not need to fiddle with memory management, making your code concise (less to write, to achieve more). The first thing you need to know about `Mat` is that you no longer need to manually allocate its memory and release it as soon as you do not need it. While doing this is still a possibility, most of the OpenCV functions will allocate its output data automatically. As a nice bonus if you pass on an already existing `Mat` object, which has already allocated the required space for the matrix, this will be reused. In other words we use at all times only as much memory as we need to perform the task. `Mat` is basically a class with two data parts: the matrix header (containing information such as the size of the matrix, the method used for storing, at which address is the matrix stored, and so on) and a pointer to the matrix containing the pixel values (taking any dimensionality depending on the method chosen for storing). The matrix header size is constant, however the size of the

matrix itself may vary from image to image and usually is larger by orders of magnitude.

Dada la expansion en el uso de las arquitecturas GPU, se comenzó a implementar un modulo adicional que contiene optimizaciones realizadas sobre GPU. El modulo fue lanzado en el 2011, contiene algunos algoritmos que ya estan implementadas en diversos modulos de OpenCV y que fueron reimplementados con el fin de obtener una aceleracion extra mediante esta arquitectura.

El manejo de estructuras de datos es importante cuando interviene codigo sobre la gpu ya que la transferencia es una parte relevante.

Por su parte, el modulo gpu define la siguiente clase:

class gpu::GpuMat Base storage class for GPU memory with reference counting. Its interface matches the Mat interface with the following limitations: no arbitrary dimensions support (only 2D) no functions that return references to their data (because references on GPU are not valid for CPU) no expression templates technique support

All GPU functions receive GpuMat as input and output arguments. This allows to invoke several GPU algorithms without downloading data. GPU module API interface is also kept similar with CPU interface where possible. So developers who are familiar with Opencv on CPU could start using GPU straightaway.

Como se dijo en el capitulo previo, la funcion inicial de la gpu era renderizar imagenes a partir de escenas. Con el tiempo la generalizacion en las aplicaciones llevo a que se implementen funciones totalmente distintas. Por ej. mediante el modulo de gpu de la libreria OpenCV se estan realizando la funcion opuesta, que es entender las escenas a partir de imagenes. ((ver filmina 11 de la presentacion))

Capítulo 3

Trabajo experimental

3.1. Fútbol robot

3.2. Implementacion existente

3.2.1. La libreria bottracker

3.2.2. Programa cliente

3.3. Implementaciones sobre GPU

3.3.1. Resultados

Capítulo 4

Conclusiones y trabajo futuro