

Computacion de altas prestaciones aplicada al reconocimiento facial

Bayarri Brian, Bayarri Jennifer, Guayta Pablo, Santillan Facundo

¹Universidad Nacional de La Matanza,
Departamento de Ingeniería e Investigaciones Tecnológicas,
Florencio Varela 1903 - San Justo, Argentina
bri_ezeba@hotmail.com, jennifer.yb@gmail.com, pablogabriel567@gmail.com, facundosantillan21@gmail.com

Resumen. El siguiente paper es un trabajo de investigación que tiene como objetivo abordar la temática computación de altas prestaciones y explicar de qué manera podemos combinar los beneficios del paralelismo con técnicas de procesamiento de imágenes, a través del algoritmo de reconocimiento de imágenes SURF (Speeded-Up Robust Features), para así adquirir una nueva funcionalidad dentro del proyecto FitVending. Esta investigación consiste en analizar y reconocer el rostro de un usuario que desea adquirir un producto y no cuenta con su dispositivo móvil mejorando al máximo la UX (user experience).

Palabras claves: GPU, SURF, paralelismo, reconocimiento facial, vending.

1 Introducción

Día tras día, la tecnología evoluciona a pasos agigantados y junto a ella, la sociedad, y es esta misma que demanda mayor uso de tecnología desde los aspectos más mínimos y cotidianos como calentar agua para una infusión hasta grandes proyectos tales como fotografiar un agujero negro a miles de kilómetros. A raíz de esto, surge la gran necesidad de personas trabajando en el área de la informática, investigación y desarrollo de nuevas tecnologías dando lugar a una problemática que muchas personas desconocen y se ven perjudicadas, el sedentarismo.

En busca de una solución a este problema surge el proyecto FitVending. El mismo consiste en una máquina vending que expende productos elegidos por el cliente a través de una aplicación móvil. Dicho proyecto presenta un punto débil y es que el usuario debe tener consigo mismo un dispositivo móvil para reclamar su producto y es aquí donde introducimos el reconocimiento facial mediante el cual la máquina vending adquiere toda la información del usuario, incluyendo sus puntos disponibles, para que de esta forma pueda cumplir su deseo y obtener su producto. Existen diversos algoritmos y mecanismos de reconocimiento facial tales como BRIEF, ORB BRISK, SURF, SIFT, entre otros, pero nosotros nos centraremos en SURF de OpenCV debido a que es utilizado en GPU dándonos la posibilidad de conseguir un mayor rendimiento en cálculos de procesamiento de imágenes sumado a que tiene un coste computacional menor en comparación con los otros algoritmos.

2 Desarrollo

Nuestra investigación surge como ampliación del proyecto FitVending. Actualmente el mismo expende productos saludables elegidos por el usuario a través de una aplicación móvil pero con la peculiaridad de que el pago no se realiza con dinero sino que con puntos, por lo que el objetivo de utilizar paralelismo está vinculado con la necesidad de darle al usuario la posibilidad de elegir el producto a través del reconocimiento facial. Dicho reconocimiento lo realizaremos mediante el algoritmo de procesamiento de imágenes SURF de OpenCV explotando el paralelismo. Para ello, agregaremos una cámara 4K al sistema embebido, una GPU NVIDIA con arquitectura Kepler, botones para elegir producto y una base de datos SQL Server remota, estableciendo comunicación con protocolo SFTP, la cual contendrá los datos del cliente, incluyendo su rostro. Por lo que es obligatorio para usar la aplicación que el usuario se registre dejando sus datos personales y físicos tales como

edad, peso, altura como tambien un escaneo de su rostro con la camara del dispositivo. En caso de no poseer una camara para escanear su rostro, el cliente no podra utilizar esta funcionalidad hasta que la registre por otro medio.

La arquitectura sera similar a la siguiente:



3 Explicación del algoritmo.

¿Cómo funciona SURF? Este algoritmo utiliza una matriz Hessiana obteniendo mayor precisión y rapidez en los cálculos por lo que, en primera instancia calcula la imagen con dicha matriz y obtiene los puntos mas característicos y localizaciones de interés, posteriormente se obtiene el patrón con el cual comparar para finalmente buscar coincidencias. Llevándolo a nuestro proyecto, en un primer paso la vending captará el rostro del usuario siendo este el patrón a utilizar por SURF como también los rostros registrados en la base de datos (existirá una lógica de programación para un filtrado en los rostros a comparar). Luego de la obtención de la imagen, se reservará memoria en la GPU para alojar el patrón, las imágenes en las cuales se buscará coincidencias y el algoritmo a ejecutar. Para el procesamiento de fotos, sabemos que una foto 4K en promedio tiene una resolución de 3840×2160 pixeles, es decir, 8.294.440 pixeles, y, por otro lado, una foto promedio de 12 MP (la registrada por el usuario), tiene una resolución de 1600×1200 pixeles, es decir, 1.920.000 pixeles. Si sumamos todos los pixeles totales a procesar y utilizamos un hilo por pixel, deberemos utilizar 10.214.400 hilos. El hardware GPU que utilizaremos nos limita a una administración de bloques-hilos en tres dimensiones (64, 64, 16), y una administración grilla-bloque en tres dimensiones (512, 512, 128). Bajo estas limitaciones, cada bloque podrá alojar 65.536 hilos, por lo que, al necesitar utilizar 10.214.400 hilos, requerimos de 156 bloques y una grilla. Ya en la GPU, y con una correcta administración de grillas-bloques-hilos, el algoritmo buscara coincidencias entre el rostro reconocido por la vending y los existentes en la base de datos. Finalmente, el resultado será alojado en memoria y captado por la unidad de procesamiento y en caso de coincidencia o un alto nivel de coincidencia, se solicita la información del usuario, caso contrario, no se podrá procesar el pedido.

Pseudocódigo ejemplo de la parte secuencial:

```
incluir OpenCV.Surf
function main()
    patron = function sacarFoto()
    listaRostros = function obtenerRostros(bd_vending, SFTP)
    function reservarMemoriaGpu()
        .....
    function copiarDatosAGpu(patron, listaRostros, SURF)
        .....
    function ejecutarAlgoritmo()
        .....
    resultado = function obtenerResultado()
    function liberarGPU()
    si resultado.valor == true
        datosUsuario = function obtenerDatos(bd_vending, SFTP, resultado.usuario)
    sino
        function informarError()
    return 0
```

Pseudocódigo ejemplo de la realización de paralelismo:

```
function ejecutarAlgoritmo()
    iterar desde 0 hasta CANTIDAD_THREADS en i:
        thread[i].start(ejecutarSurf())
    join.threads()
```

Pseudocódigo ejemplo del algoritmo SURF:

```
//Obtenemos rostro escaneado por la vending
patron = memGlobal.read("rostro.png")
//Obtenemos un rostro registrado por el usuario
rostro = memGlobal.read[rostros[iterado]]
//Creamos un objeto SURF
surf = new SURF()
//Buscamos puntos de interes en las imagenes
interesP = surf.detectAndCompute(patron)
interesR = surf.detectAndCompute(rostro)
//Dibujamos los puntos de interes encontrados
surf.drawnKeyPoints(patron)
surf.drawnKeyPoints(rostro)
//Por ultimo, realizamos la comparacion
resultado = surf.compare(patron, rostro)
return resultado
```

4 Pruebas que pueden realizarse

Una vez implementado el sistema FitVending con sus correspondientes agregados, recomendamos realizar pruebas con una gran variedad de rostros y distinta información de usuario, variando luz de ambiente, distancia con la cámara, posición con respecto al foco, etc. Sumado a esto, consideramos también verificar el correcto

descuento de puntos o monedas al solicitar un producto para confirmar que el sistema se encuentra estable, robusto y con un correcto funcionamiento.

5 Conclusiones

Este trabajo nos deja una grata introduccion de como podemos aprovechar distintas tecnologias que podemos integrar a nuestros proyectos, las cuales son muy optimas y pueden funcionar en cualquier entorno de trabajo debido a lo adaptable que son los algoritmos de paralelismo. En nuestro caso fue el algoritmo SURF para procesamiento de imágenes, pero hay un mundo de algoritmos para distintos fines y proyectos.

Consideramos nuestra solucion como innovadora por el impacto social que y la excelente UX que puede generar como tambien ser el punta pie inicial para grandes proyectos y tecnologias que lleven a la sociedad a un futuro mejor.

6 Referencias

1. Ebrahim Karami, Siva Prasad, Mohamed Shehata: Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison. St Johns, Canada (2015).
2. Zinelabidine Boulkenafet, Jukka Komulainen: Face Antispoofing Using Speeded-Up Robust Features (2016)
3. Benavidez Alvares, Roman Alonso, Avilez Cruz: Identificacion de rostros por tecnica de puntos de interes SURF. Instituto tecnologico de Celaya, Mexico (2015).
4. NVIDIA's Next Generation CUDA Compute Architecture: Kepler TM GK110/210. (2017)
5. Cheon, S., Eom, I.K., Ha. : An enhanced SURF algorithm base on new interes point detecion procedure and fast computation technique (2015)