

Reconocimiento canino por imagen

Calcagno Leandro, Cuesta Nicolás, Martín Mora Juan Manuel

Universidad Nacional de La Matanza,
Departamento de Ingeniería e Investigaciones Tecnológicas,
Florencio Varela 1903 - San Justo, Argentina
lcalcagno@alumno.unlam.edu.ar
ncuesta@alumno.unlam.edu.ar
jmartinmora@alumno.unlam.edu.ar

Resumen. La presente investigación tiene como finalidad sumergirse en el mundo del reconocimiento de imágenes a través de patrones, para poder detectar mediante el algoritmo SURF (Speeded-Up Robust Features) animales domésticos y que el mismo se complemente con el proyecto “collar inteligente”, el cual fue desarrollado por nuestro equipo. De esta manera, suma funcionalidad al producto permitiendo identificar si en efecto ese animal le pertenece al dueño o no, para poder hacer uso de las demás funcionalidades que ya provee el Software. El algoritmo que se desarrolla en este documento forma parte de una de las ramas de la ciencia computacional, la inteligencia artificial.

Palabras claves: SURF, GPU, CUDA.

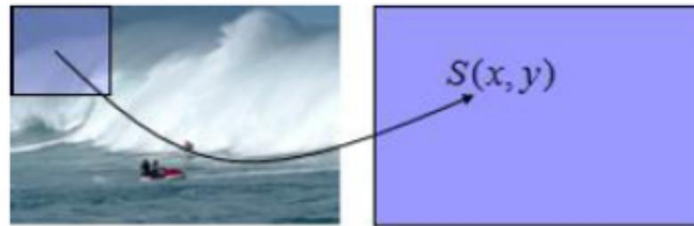
1 Introducción

La sociedad y la tecnología van evolucionando día a día, y es la misma sociedad la que demanda el mayor uso de tecnologías en aspectos mínimos pero que no dejan de ser importante cotidianamente. Como consecuencia, se decide investigar el reconocimiento de imágenes en dispositivos móviles para complementar el proyecto del collar inteligente de un animal doméstico.

De esta manera se puede identificar si el animal fotografiado por la cámara, es el que le pertenece al dueño o no, y el mismo podrá decidir que acción tomar en caso de ser positivo.

En la actualidad existen diversos métodos usados en el reconocimiento de imágenes, algunos con mayor o menor precisión. Entre ellos, se encuentran los algoritmos SIFT, SURF (implementado a partir de SIFT), BRIEF, entre otros. De todos ellos, hemos elegido el algoritmo SURF por las siguientes razones:

- SURF ofrece la detección de objetos con menos coste computacional, pero sin embargo se requieren de un alto nivel de procesamiento de información [4].
- El método surf es utilizado en GPU, las cuales tienen la tecnología para conseguir un mayor rendimiento o velocidad en cálculos de procesamiento, trabaja incluso con imágenes HD. [1]
- Otra ventaja de SURF es que utiliza cuadrados para promediar la zona de la imagen, utilizando para ello la Imagen integral descrita en la siguiente ecuación:

$$S(x, y) = \sum_{x' \leq x} \sum_{y' \leq y} I(x', y')$$


siendo $S(x, y)$ la imagen integral y $I(x', y')$ la imagen original, luego se procede a realizar una convolución entre la imagen integral y el filtro Haar para encontrar la suma. [3]

- El algoritmo hace uso de la matriz Hessiana $H(x, \sigma)$ la cual permite mayor precisión y rapidez en los cálculos [3].
- Es compatible con OpenCV (librería CUDA-SURF).

Finalmente, la tecnología CUDA hace alusión a un compilador así como a un conjunto de herramientas de desarrollo, que permite a los programadores usar una variación del lenguaje de programación C para codificar algoritmos en GPU, y de esta forma aprovechar su gran potencia e incrementar el rendimiento del sistema.

2 Desarrollo

Para llevar a cabo la programación de la funcionalidad, se eligieron los siguientes lenguajes, herramientas y librerías:

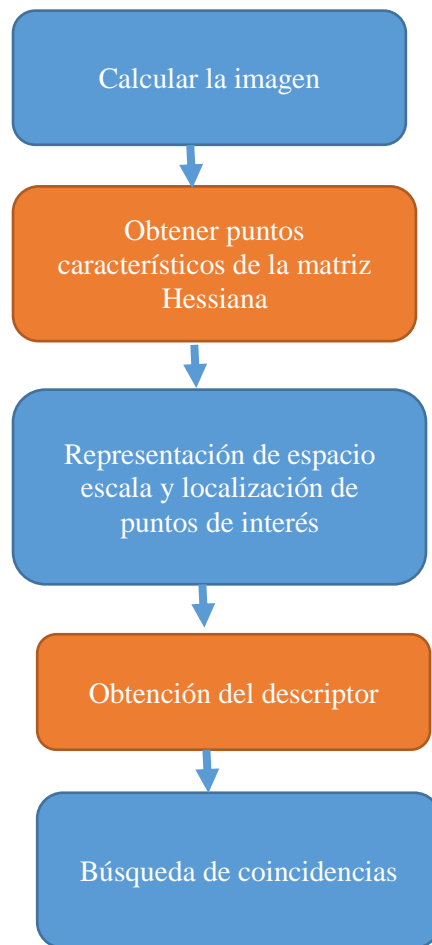
- **Android NDK:** Con el fin de poder incorporar nuestra funcionalidad a la aplicación Android, es necesario hacer uso del Android NDK. El mismo es un conjunto de herramientas que le permite implementar partes de su aplicación en código nativo, utilizando lenguajes como C y C++.
- **Lenguajes C/C++:** Lenguajes que permiten hacer uso de las librerías de Open CV y ejecutarse en Android de forma nativa.
- **Librería CUDA-SURF de OPEN CV:** Utilizado para la generación del algoritmo SURF en un entorno que permita la ejecución en paralelo.
- **Sistema Operativo Android:** Para el desarrollo se utilizara Android 5.0 (nivel 21 de Android API).

La funcionalidad solamente se encontrará disponible en modelos de celulares que cuenten con GPU incorporada, dejando excluidas por el momento a todos los modelos que no cuenten con la misma. A futuro se podrá llevar a cabo una investigación para que el procesamiento de imágenes sea en servidores y el celular reciba directamente la notificación.

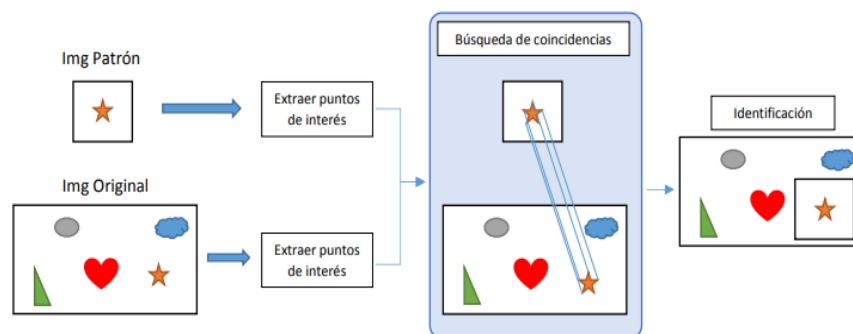
3 Explicación del algoritmo.

Idea del desarrollo SURF

El algoritmo cuenta con 5 etapas mostradas a continuación:



[1]



Ejecución del programa

El primer paso será obtener una imagen del can. Luego de la obtención de la imagen, se deberá reservar memoria en la GPU para transferir tanto la imagen como el algoritmo a ejecutar, en este caso SURF. Se aprovechará el beneficio de dicho algoritmo (trabajar por regiones y no por píxeles) para el diseño de ejecución (thread, Bloques y Grillas). Cada región será responsabilidad de un thread. La funcionalidad considera un promedio de resolución de imagen de 12MP. Esto es un total de 37.748.736 de píxeles. Si cada región tiene 3072 píxeles, entonces serán

necesarios 1024 threads para su ejecución. Los mismos serán distribuidos en 16 bloques siendo que cada bloque soporta hasta 64 threads. Una vez realizado el procesamiento, se debe transferir el resultado desde la GPU a la aplicación. Dicho resultado informara el grado de coincidencia que existe entre la imagen tomada por la cámara y la imagen almacenada, valuada en porcentaje.

```
int main(){
    static float porcentaje_acierto = 0.9;
    float resultado = 0.0;
    //Obtener la imagen de la camara
    obtener_imagen();
    //Subdividir la imagen en regiones
    imagen[][] = imagen_a_regiones();
    //Reservar memoria en gpu con cudaMalloc();
    reservar_memoria_gpu();
    //Copiar la imagen al gpu con cudaMemcpy(*a,*b, tam, cudaMemcpyHostToDevice);
    copiar_regiones_a_gpu();
    //Ejecucion de algoritmo SURF en GPU
    ejecutarSURFEnGPU <<<16,64>>> imagen,resultado;
    //Para obtener el resultado. Varía del anterior por la palabra cudaMemcpyDeviceToHost.
    resultado = copiar_gpu_a_cpu()
    //En caso de ser positivo, se ejecuta la acción programada.
    if(resultado >= porcentaje_acierto){
        ejecutar_accion();
    }
    //Finalmente se libera la memoria de la GPU
    liberar_memoria_gpu();
    return 0;
}

_global_ ejecutarSURFEnGPU(){

};
```

4 Pruebas que pueden realizarse

Una vez desarrollado el algoritmo SURF y su implementación mediante GPU, la aplicación del mismo brindará un plus funcional al proyecto ya desarrollado del collar inteligente para un animal doméstico. Esta investigación presentaría la oportunidad de reconocer si la mascota que se aproxima a una abertura es la que le pertenece al dueño, quien previamente en su registración de la app cargaría fotos de sus mascotas. Al disponer cámaras IP cerca del perímetro de las aberturas, el sistema captaría fotos del animal y las compararía con las cargadas en la app, permitiendo el ingreso del animal a la vivienda en caso de que haya sido configurado como automático, o consultando al dueño si desea que su mascota ingrese a su hogar.

Actualmente lo consideramos un proyecto innovador a lo que se conoce hoy en día.

5 Conclusiones

Gracias a esta investigación del funcionamiento del algoritmo SURF para el procesamiento de imágenes; cuya ejecución será llevada a cabo con GPU utilizando CUDA para el diseño de la solución, supimos comprender la complejidad del algoritmo que se debe aplicar a una imagen para que un procesador logre devolver el resultado de lo solicitado en tiempo y forma. No es mínimo resaltar que para que esta implementación funcione, se necesita un tiempo de respuesta alto y con precisión, ya

que el animal podría alejarse rápidamente o chocarse con la abertura en caso de que quiera ingresar.

Esta investigación abrirá puertas para seguir desarrollando este proyecto que va mutando y avanzando cada vez más, con el objetivo brindarle al usuario las mejores soluciones a su vida cotidiana.

6 Referencias

1. Navacerrada, J.: Sistema de identificación de matrículas con OpenCV. Universidad politécnica, Madrid (2017).
2. Alcaraz Chávez, J., Calderón Solorio F., Implementación en tiempo real para el seguimiento en secuencias de video. ReCIBE (2014).
3. Benavidez Alvarez C., Roman Alonso G, Villegas Cortes J, Aviles Cruz, C.: Identificación de rostros por técnica de puntos de interés SURF. Instituto tecnológico de Celaya, Méjico (2015).
4. Cheon, S., Eom, I.K., Ha, S.W., Moon, Y.H.: An enhanced SURF algorithm based on new interest point detection procedure and fast computation technique. J. Real-Time Image