

# Reconocimiento canino por imagen

Calcagno Leandro, Cuesta Nicolás, Martín Mora Juan Manuel

<sup>1</sup>Universidad Nacional de La Matanza,  
Departamento de Ingeniería e Investigaciones Tecnológicas,  
Florencio Varela 1903 - San Justo, Argentina

[lcalcagno@alumno.unlam.edu.ar](mailto:lcalcagno@alumno.unlam.edu.ar)

[ncuesta@alumno.unlam.edu.ar](mailto:ncuesta@alumno.unlam.edu.ar)

[jmartinmora@alumno.unlam.edu.ar](mailto: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. SURF es un algoritmo de visión por computador, capaz de obtener una representación visual de una imagen y extraer una información detallada y específica del contenido. Fue planteado por Herbert Bay en el año 2006, y este algoritmo hace réplicas de la imagen para así buscar los puntos que estén en todas las réplicas asegurando la invariancia de escala. Las réplicas pueden ser piramidales o del mismo tamaño a la original. En las piramidales la principal idea es tener una imagen más pequeña que la anterior, para ello se realiza un filtro de la imagen y se submuestra ya que si se realiza solo el submuestreo se pierde información de la imagen.

SURF ofrece la detección de objetos con menos coste computacional, pero sin embargo se requieren de un alto nivel de procesamiento de información. 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.

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

En SURF, para identificar puntos de interés tales como esquinas, los detectores más usados son el Harris, y el Hessiano. A lo largo de varios estudios se determinó que el detector Hessiano basado en el determinante de la matriz hessiana con el Laplaciano era más estable, robusto e invariante a cambios de escala que el detector de Harris. Un punto de interés es una parte de la imagen con características destacables. Habitualmente son bordes, esquinas o formas de T, que hacen que sean puntos muy exclusivos.[1]

### *Detección de puntos de interés*

La matriz Hessiana  $H(x, \sigma)$  permite mayor precisión y rapidez en los cálculos, cuya ecuación es:

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix}$$

teniendo en cuenta que la derivada de segundo orden del filtro de Gauss es:

$$\frac{\partial^2}{\partial x^2} g(\sigma)$$

se realiza una convolución con la imagen  $I$  en el punto  $x = (x, y)$  para obtener el valor de  $L_{xx}(x, \sigma)$ ; de igual manera para obtener  $L_{xy}(x, \sigma)$  y  $L_{yy}(x, \sigma)$  el proceso es similar, estas derivadas son conocidas como derivadas Laplacianas de Gausianas.

$$\begin{aligned} L_{xx}(x, y, \sigma) &= \frac{\partial^2}{\partial x^2} * (G(x, y, \sigma) * I(x, y)) \\ &= \left( \frac{\partial^2}{\partial x^2} * (G(x, y, \sigma)) \right) * I(x, y) \\ L_{xx}(x, y, \sigma) &= G_{xx}(x, y, \sigma) * I(x, y) \end{aligned}$$

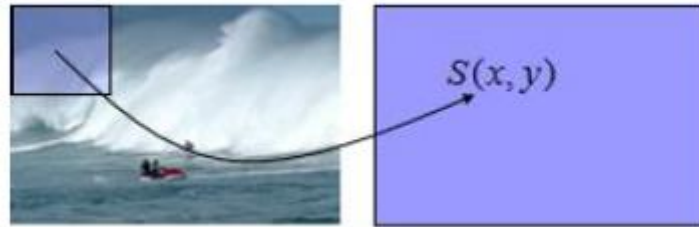
El cálculo de los determinantes de la matriz Hessiana que permite localizar la escala del punto se define a continuación:

$$\det(H_{\text{aprox}}) = D_{xx}D_{yy} - (\omega D_{xy})^2$$

donde  $D_{xx}$ ,  $D_{yy}$ ,  $D_{xy}$  son las aproximaciones de las derivadas parciales,  $\omega$  es el peso para balancear la determinante Hessiana, el cálculo para  $D_{xx}$  se realiza a partir de  $I(x, y)$  y  $G_{xx}(x, y)$ , de manera similar para  $D_{yy}$ ,  $D_{xy}$ .

La 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.

#### *Asignación de orientación*

Una vez realizada la convolución y seleccionados los puntos máximos, se debe disponer de la orientación y la descripción de la vecindad. Se establece un descriptor para cada punto con la ayuda de los filtros Haar que nos brindan una solución en los ejes  $x$  y  $y$  en una región circular de 6s usado por el autor, siendo  $s$  la escala del punto de interés.

Con los cálculos realizados para todos los vecinos se estima la orientación dominante, se procede a sumar todos los resultados dentro de una ventana deslizante que cubre un ángulo de  $\pi/3$ .

#### *Creación del descriptor*

Se construye como primer paso una región cuadrada de tamaño  $20s$  alrededor del punto de interés y orientada en relación a la orientación calculada en la etapa anterior[2]. La región es dividida en subregiones de  $4 \times 4$ , en cada una de las cuales se calculan las respuestas de Haar siendo estas  $d_x$  y  $d_y$ , para cada subregión se suman los resultados así como sus valores absolutos  $|d_x|$ ,  $|d_y|$  como se muestra en la figura 4, quedando formado un vector para cada una de las regiones presentado en la siguiente ecuación, los puntos resultantes son invariantes a la escala.

$$v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$$

De esta forma se obtiene un descriptor de 64 valores para cada punto identificado.

Los factores principales que se deben tener en consideración en la detección de rostros son: (i) iluminación, (ii) orientación, (iii) escala y (iv) textura. El análisis automatizado del contenido de una imagen se refiere a la detección y estudio de su contenido como ser los colores, formas, texturas o cualquier otra información que pueda derivarse de la propia imagen. Específicamente para la detección de rostros la metodología tradicional marca 4 etapas: (i) detección de la cara, (ii) alineación de la cara, (iii) extracción de las características y (iv) reconocimiento [3].

### 3 Explicación del algoritmo.

A partir de la obtención de la imagen, el algoritmo deberá llevar a cabo la separación en regiones, para así llevar a cabo las operaciones necesarias a ejecutar en SURF en cada una de ellas en la GPU y luego verificar el grado de coincidencia respecto a una imagen almacenada. El diseño de ejecución en la GPU (Threads, Bloques y Grillas) dependerá del tamaño considerado para las regiones. Una vez realizado el procesamiento, se debe buscar el resultado a la GPU para luego tomar una decisión sobre el grado de coincidencia.

```
int main(){

    //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 <<<<cantBloques,cantThreads>>>> 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 == verdadero){
        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 actualmente.

## **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, México (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 Process (2016)