Procesamiento HPC en Sistema de Reciclaje inteligente

Facal Ernesto
Oria Joaquín Ariel
Marsón Tomás
Schiro Andrés

Universidad Nacional de La Matanza,
Departamento de Ingeniería e Investigaciones Tecnológicas,
Florencio Varela 1903 - San Justo, Argentina

Resumen. El objetivo de esta investigación se basa en el análisis, manipulación y reconocimiento de imágenes para un sistema reciclador de basura integrado. Se utiliza un algoritmo para la detección de formas de imágenes haciendo uso de la Distancia de Hausdorff y se presenta una implementación en forma paralela utilizando OpenMP.

Palabras claves: OpenMP, Paralelismo, Hausdorff

Introducción

La investigación que se desarrolla en este documento está aplicada a un Sistema de reciclaje de materiales inteligente integrado. Este sistema cuenta con una cámara que monitorea el recipiente donde se colocan los materiales, dicha cámara proveerá al sistema las imágenes a procesar. Una vez identificado el material el sistema procederá a ubicar el material en el contenedor indicado según su clasificación.

Para solucionar el problema de reconocimiento de imágenes o de elementos en una imagen de forma eficiente se propone utilizar la Distancia de Hausdorff.

Este concepto mide el nivel de discrepancia entre dos grupos de puntos de un mismo espacio métrico. Este algoritmo propuesto ofrece la posibilidad de paralelizar la parte más pesada de la computación.

Dado que para obtener una aproximación fiable acerca del tipo de material debemos compararlo con un número considerable de imágenes distintas de ese tipo, va a haber un gran procesamiento de información para cada comparación antes de llegar a un resultado final. En esta etapa comienza a surgir la idea de paralelizar estas tareas.

Como herramienta de paralelización se ha decidido usar OpenMP porque ofrece una interfaz sencilla que permite realizar la paralelización introduciendo pocos cambios sobre la implementación secuencial.

Desarrollo

El sistema de reciclaje inteligente cuenta con una plataforma conectada a un sistema de volcado que utiliza servomotores para controlar los ejes de rotación y giro de la plataforma. Los residuos que son situados en la plataforma serán colocados en contenedores específicos según su clasificación. Para realizar esta clasificación el sistema contara con una cámara de manera tal que, se identifique de forma gráfica el tipo de residuo, comparando la foto del residuo colocado en la plataforma con las imágenes almacenadas en una base de datos. Los residuos podrán ser clasificados en "Papel", "Plástico", "Vidrio", "Metal" o "Orgánico". Para realizar la comparación de imágenes y obtener la clasificación del tipo de residuo utilizaremos el método de la Distancia Hausdorff. El procesamiento de la imagen será realizado por un sistema de cómputo aparte que se comunicará con la placa Arduino. La placa Arduino indica el comienzo del proceso una vez que detecta un residuo en la plataforma a través de un sensor de peso y luego recibe el resultado para finalizar la clasificación.

La idea tras la distancia Hausdorff es que puede determinarse una distancia d que indica el grado de disparidad entre conjuntos de puntos pertenecientes al mismo espacio. Si la distancia vale 0 ambos grupos serán el mismo, y cuanto mayor sea el valor, más diferentes entre sí. Esta propiedad puede usarse como base para el análisis de imágenes en formato mapa de bits, ya que, en este caso, las coordenadas de los pixeles forman los elementos del espacio y es posible comparar grupos que pertenezcan a éste. Es común aplicar la distancia de Hausdorff para obtener la medida de similitud entre dos contornos.

La definición formal de la distancia Hausdorff viene dada por la siguiente expresión matemática:

Sean $A = \{a_1, ..., a_p\}$ y $B = \{b_1, ..., b_q\}$ dos conjuntos finitos no vacíos, subconjuntos de un espacio métrico (M, d), se define la distancia Hausdorff $d_H(A, B)$

$$d_H(A, B) = max\{h(A, B), h(B, A)\}$$

Donde

$$h(A, B) = max\, min||a - b||$$

$$h(B, A) = max\, min||b - a||$$

Con $a \in A$, $b \in B$ y $\|\cdot\|$ es una norma entre los puntos de A y B.

Para la paralelización de este algoritmo se utiliza OpenMP, una API ideada para la programación de tareas multiproceso con memoria compartida. La API consiste en una agrupación de directivas para el compilador, rutinas de biblioteca y variables de entorno que permiten modificar el comportamiento de la ejecución de un programa. El funcionamiento básico de OpenMP, se basa en la idea de fork-join, donde el hilo de ejecución maestro se divide en diversos hilos.

Explicación del algoritmo

La placa Arduino envía una señal al sistema de cómputo indicando que hay un residuo en la plataforma para el comienzo del procesamiento.

Parte secuencial: Se envía la foto sacada por el sistema de reciclaje al sistema de cómputo para que analice la imagen y retorne el resultado. Se enviará una foto por cada residuo ingresado.

Parte paralela: Aquí es donde se compara la imagen buscando coincidencias contra las imágenes almacenadas en la base de datos del sistema de cómputo.

Pseudocódigo

```
// se obtiene la imagen a procesar
Imagen = ObtenerImagen()

// se realiza la comparación (parte paralelizable)
Resultado = DistanciaHausdorff(Imagen)

// se clasifica el resultado obtenido
TipoResiduo = ClasificarSegunTipoResiduo(Resultado)

// envía a la placa Arduino el tipo de residuo para que proceda a reciclarlo según corresponda
Respuesta (TipoResiduo)
```

A continuación, se explicará la función DistanciaHausdorff.

La función consiste en tres bucles anidados, el primer for recorre las imágenes almacenadas en la base de datos, las cuales ya se encuentran expresadas en contorno, el cual está almacenado en un vector de coordenadas. El segundo recorre los pixeles

de los contornos de la imagen que se obtuvo del residuo de la plataforma y en el tercer for, para cada píxel, recorre todos los pixeles de los contornos de la primera imagen y calculara las distancias entre estos y se guardara siempre la menor. Al finalizar el recorrido para un punto dado de la primera imagen seleccionará aquella distancia que sea mayor entre la nueva y el máximo anterior. Este proceso se realiza en el sentido imagen1 → imagen2 y luego en el sentido imagen1 ← imagen2. Finalmente devolverá la imagen que tenga menos distancia con la imagen original.

DistanciaHausdorff (Imagen)

```
//Para cada imagen de la BD…
For(  Imágenes BD[ImagenN] ){
   // Para cada Pixel de la imagen Residuo
   For ( Imagen del Residuo[PixelN] )
        // Para cada Pixel de la imagen BD…
       For (ImágenesBD[ImagenN[PixelN]]){
          Obtener distancia mínima ()
            }
   Obtener distancia máxima ()
 }
}
Obtener imagen más similar ();
// Retorno la imagen con menos distancia entre la original para que se
clasifique posteriormente
Return imagen más similar;
```

Pruebas que pueden realizarse

Una de las pruebas que pueden realizarse es implementar el algoritmo con y sin el empleo de OpenMP, para así demostrar la diferencia en los tiempos de ejecución entre la versión secuencial y la optimizada paralelizable con OpenMP.

Los casos de prueba que se pueden realizar son colocando residuos en la plataforma del sistema de reciclaje, previamente habiendo cargado imágenes en la base de datos de ese material para que el sistema pueda comparar y realizar el procesamiento.

Conclusiones

Podemos ver que a través del uso de la propiedad de la distancia de Hausdorff es posible realizar un reconocimiento de patrones en imágenes y de esta forma poder identificar los distintos tipos de materiales para que luego el sistema integrado una vez ya clasificado el material lo recicle según corresponda.

Se utilizó OpenMP para ejecutar el algoritmo comparando paralelamente la imagen del objeto con las de la base de datos y así aumentar notablemente su rendimiento.

Referencias

1. OpenMP versus Threading in C/C++ (2014)

2. Measuring synchronization and cheduling overheads in openMp (2016)

3. Paralel computing using openMp (2017)

4. Comparación de Huellas Dactilares Usando la Distancia Hausdorff (2008)