

RECONOCIMIENTO DE PATRONES FACIALES EN TIEMPO REAL MEDIANTE TRANSFORMADA DE WAVELET Y COMPUTACIÓN PARALELA

AUTORES:

Pablo Crovetto, jcrovetto@uni.pe
Daniel Palomino, dpalominop@uni.pe
Santiago Cortijo, scortijoa@uni.pe

ASESOR:

Dr. Javier Solano, jsolano@uni.edu.pe

Centro de Tecnologías de Información y Comunicaciones **UNIVERSIDAD NACIONAL DE INGENIERÍA**

Resumen— Este proyecto aborda el problema de detección y reconocimiento de rostros en tiempo real utilizando un ordenador y de manera autónoma.

Se utiliza una base de datos con muchas imágenes de cada individuo a reconocerse.

La primera fase de este proyecto consiste en implementar un detector de piel, basado en la compresión de imágenes mediante la transformada de wavelet-Daubechies, con el fin de obtener sus bajas frecuencias de intensidades de color, para así evaluarlas en la RN perceptrón y poder detectar piel humana.

La piel detectada nos da nuestra región de interés, en la que se ha de ubicar nuestro rostro, y sobre ésta se calculan los parámetros faciales que identifiquen a cada rostro de manera única.

Finalmente, se utiliza una RN de tipo Kohonen, cuya capa de entrada es un conjunto de parámetros faciales invariantes (rotacionales y escalares) los cuales fueron obtenidos en el paso anterior, y cuya finalidad es relacionar los datos de la capa de entrada con los de nuestra base de datos y determinar a que rostro es el que corresponde.

Para lograr implementar la transformada de wavelet-Daubechies en tiempo real se utiliza un paradigma de programación paralela (GPGPU General Purpose Graphic Processing Unit).

Palabras clave— Wavelet 2D, Detector de Piel, Visión Computacional, Redes Neuronales, GPGPU

Abstract Abstract— A well known problem in human-computer interaction is detection and face recognition in real time.

A database with many pictures of each individual to be recognized is taken.

The first phase of this project is to implement a skin detector based on image compression using Daubechies wavelet-transform in order to get their low frequencies of color intensities and the we can evaluate them in the RN perceptron so we can detect human skin.

The detected skin gives our region of interest, and the face will be located in this region, and over this area facial parameters that identify uniquely each face will be

calculated. Then, a Kohonen NN is trained and his input layer is an invariant (rotational and scalar) set of facial parameters, this parameters were obtained from previes step, and so the goal of this RN is to relay the paramenters input with our database and the we can link the recognized face with someone in the database saved originally if it is the case.

For successfully implement the Daubechies wavelet-transform in real time a parallel programming paradigm (GPGPU: General Purpose Graphic Processing Unit) is used.

Key words: Wavelet 2D, Skin detection, Computer Vision, Neural Network, GPGPU.

1. Introducción

El hecho de contar con tecnologías nuevas para el procesamiento de datos a mayor escala y cada vez más rápido permite la investigación hacia nuevos horizontes. Siendo así, hemos utilizado un paradigma de programación paralelo que usa una GPU para hacer los cálculos, aprovechando la gran cantidad de procesadores que dispone.

Se distingue dos temas diferenciados en nuestro proyecto; es decir, nos encontramos con dos algoritmos críticos, por un lado encontrar un rostro y por otro identificarlo. En esta primera parte hemos identificado los algoritmos con mejores prestaciones para ser paralelizados, con esto hemos conseguido no aún mejores resultados, pero sí la reducción de tiempos de ejecución. Lo cual contribuye a desarrollar un sistema de reconocimiento de rostro bajo condiciones controladas en tiempo real.

Comprender e igualar el sistema de reconocimiento del ser humano es una tarea que aún no se ha logrado, aunque muchos métodos han sido diseñados y desarrollados para tal propósito con resultados cada vez más exactos, no se ha podido obtener un método que abarque todos los casos posibles. Pues las clases de rostros exhiben una variabilidad intrínseca que es difícil sino imposible de caracterizar analíticamente[7]. Algunos de estos son las formas, tamaños, edades, género, estilo de cabello y expresiones faciales, además de condiciones extrínsecas que dificultan aún más la tarea como maquillaje, lentes, la fuente de iluminación, la presencia de otros objetos, el

mismo sistema de imágenes, etc. Pero la fuente que sin duda causa más problemas es el cambio de posición. Por tanto nuestro sistema solo tratará de identificar a agentes que están en una vista frontal a la cámara.

Para detectar un rostro hay en esencia dos categorías[6,7,8] una que es por extracción de características y la otra que es con una plantilla holística.

La primera se basa en identificar las características más resaltantes en un rostro, así que cuando encontramos estas característica juntas en una imagen podemos afirmar que hay un rostro. Pero la mayor dificultad en esto es que debemos tener un detector de ojos, de nariz, de boca; luego el problema se ha dividido en otros pero con la misma dificultad inicial. Aunque debemos señalar que las ventajas de este tipo de enfoques se dan en la identificación del rostro. Para esto se emplearía un grafo uniendo todos los rasgos más significativos. Luego este grafo se compara con otros ya almacenados y si encaja tendríamos identificado al rostro con cierto rendimiento.

Por otro lado tenemos la detección mediante plantillas holísticas. En este caso algunas variaciones en el rostro son delineadas con pre-procesamiento o algún otro método, luego se aprende algún modelo de rostro y se busca en la imagen para ver si alguna región encaja con el modelo, y de encajar se tendría un rostro. O alternativamente se puede aprender una función de clasificación para que discrimine entre rostros y no rostros. Este método se caracteriza por usar imágenes de baja resolución, aunque no se puede aplicar cuando hay obstáculos entre la cámara y el rostro.

Así entonces escogemos el segundo método para la detección del rostro. Pues, el primero necesita de múltiples algoritmos aplicados en serie para encontrar un rostro siendo poco atractiva la paralelización, además hace uso de un grafo para modelar el rostro, que es ideal para implementarlo convencionalmente, es decir, en forma serial. Por otro lado el otro método emplea técnicas de procesamiento de imágenes, y éstas son básicamente aplicación de la transformada de Fourier o cualquier otra Transformada entre un kernel y la imagen que es una matriz. Y sabemos que toda operación con matrices es ideal para paralelizar.

Un breve repaso de Wavelets 2D

Al igual que la transformada de fourier que descompone una señal cualquiera (que cumpla con sus condiciones) en función de senos y cosenos de frecuencias múltiples de la original, la transformada de wavelet descompone la señal en función de traslaciones y escalamientos de otra onda que cumpla con ciertas condiciones de admisibilidad, tales como valor medio cero, energía finita ,entra otras[3]; a estas funciones con estas características se les denomina wavelets madre y a la que genera, simplemente, wavelets.

La ventaja de usar las wavelets es su mayor semejanza con las señales reales, que generalmente se usan, las cuales varían de una forma ondular y que concentran gran parte de su carga de información en ciertos lugares y no de manera uniforme; en este sentido se vuelven mucho más eficientes al momento de querer extraer determinadas características de una imagen gracias a su capacidad de traslación y dilatación que permite asemejarse de una manera adecuada a las formas que queremos determinar

discriminado a las otras;es decir, si quisiéramos ubicar una figura determinada bastaría con elegir como wavelet madre una ondícula que se asemeje lo más posible a esta, y aplicarle con esta wavelet la transformada bidimensional a nuestra imagen.

$$W_u(a,b_{x_1},b_{x_2}) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} u(x_1,x_2) \cdot \psi_{a,b_{x_1},b_{x_2}}(x_1,x_2) \cdot dx_1 dx_2$$

Fig. 1. Transformada continua de wavelet bidimensional[2].

Donde: b_{x_1} y b_{x_2} son las traslaciones en las dos dimensiones y

$$\psi_{a,b_{x_1},b_{x_2}}(x_1,x_2) = \frac{1}{|a|} \psi\left(\frac{x_1 - b_{x_1}}{a}, \frac{x_2 - b_{x_2}}{a}\right)$$

Fig. 2. Traslaciones y escalamiento de una wavelet[2].

Siendo: $\Psi(x_1,x_2)$ una función wavelet madre bidimensional.

La transformada continua de wavelet-2d (cwt-2d) se puede expresar en función de 2 filtros que proviene de la ondícula elegida[2]; estos filtros son usados para implementar la cwt-2d, la cual se puede expresar como la convolución de la imagen con estos filtros de manera reiterativa.

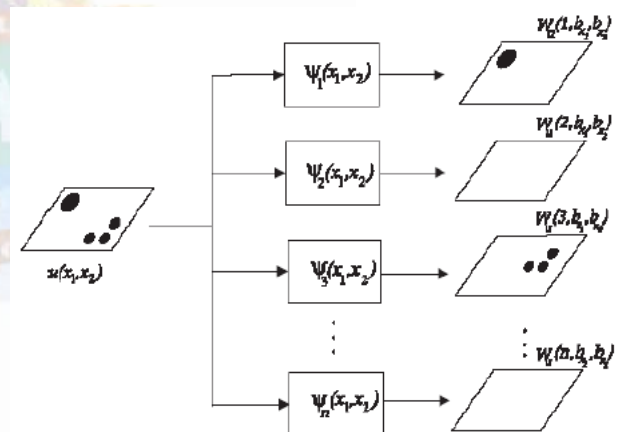


Fig. 3. Equivalente de la transformada de wavelet en forma de filtros.[2]

Dado que no se podrían calcular todas las componentes de una transformada continua, se procede a usar en la práctica la transformada discreta de wavelet-2d, la cual recoge sola las cuatro muestras más representativas.

Un breve repaso de CUDA

CUDA(Compute Unified Device Architecture) fue introducido por Nvidia como una arquitectura de cálculo en paralelo de proposito general. Que incluye tanto hardware como software(compilador de CUDA, drivers y librerías). El modelo de programación en cuda consiste en funciones que se ejecutan simultaneamente en una gran cantidad de hilos muy ligeros en la GPU. Estos hilos tiene una jerarquía: los bloques, que son un conjunto de hilos en 1, 2 o 3 dimensiones, luego esta una grid, que es un conjunto de bloques que puede ser de 1 o 2 dimensiones. Solamente los hilos en el mismo bloque pueden compartir datos y sincronizarse entre ellos. La capacidad de procesamiento, la cantidad de hilos, el número de procesadores que pueden tener varian de acuerdo al dispositivo que se tenga. En nuestro caso una Ge Force

8400, Tiene 1 streaming multiprocessor(SM), y cada multiprocessor tiene 8 streaming processors (Sps)[9].

Una tarjeta de video posee 5 tipos de memorias a las que podemos acceder, éstas son: la memoria global que es de lectura y escritura y que además posee un espacio de dirección de memoria separado para poder comunicarse con el CPU mediante el PCI bus, la memoria constante que es de solo lectura al igual que la memoria textura, la memoria compartida que es de lectura y escritura pero solo para los hilos del mismo bloque y la memoria de cada hilo, la cual obviamente sólo puede ser accesada por el hilo asignado[10].

La doble precisión es aún una debilidad pues solo algunas tarjetas tienen la característica de soportarla.

2. Materiales y métodos

En esta primera parte hemos utilizado un webcam Microsoft 360, que soporta hasta 2Mb/s de transmisión de datos. Una tarjeta de video Nvidia GeForce 8400 cuda capable.

Algoritmo para el Filtro de Piel

Entra una imagen una imagen en RGB tradicional desde la webcam, luego pasamos la imagen al espacio de color YCbCr (donde Y es la luminancia y CbCr el plano de la crominancia azul y rojo), luego hacemos una imagen binaria con los pixeles que arrojan piel en toda la imagen. Este filtro es un filtro de color calibrado manualmente sobre región del plano de crominancia CbCr; pues, la piel sobre este plano está distribuida sobre una región cerrada y acotada [5]. Así entonces calculamos un punto central de pixel y a partir de allí definimos un radio.

Algoritmo para la Transformada de Wavelets 2D

La convención usada para poder expresar la dwt-2d de una imagen es la siguiente:

- Cuadrante Arriba-Izquierda: Paso Bajo/Paso Bajo.
- Cuadrante Arriba-Derecha: Paso Bajo/Paso Alto.
- Cuadrante Abajo-Izquierda: Paso Alto/Paso Bajo.
- Cuadrante Abajo-Derecha: Paso Alto/Paso Alto.

Los algoritmos más comunes para implementar la dwt-2d son la forma estándar y la no estándar, nosotros usaremos la forma estándar por ser la más sencilla de implementar.

El algoritmo dwt-2d estándar[4] consiste en primer lugar en la convolución de cada fila con mis dos filtros (Pasa Bajos y Pasa Altos), esto dará un conjunto de vectores filas las cuales serán reducida a la mitad de su tamaño resultante mediante un proceso que se denomina diezmado, el cual consiste en la eliminación de un elemento por cada dos existentes. Una vez hecho estos se procede a unir los resultados de la convolución de cada fila con los filtros, esto dará un solo vector fila por cada uno existente inicialmente que tendrá en la primera mitad las bajas frecuencias y en la segunda, las altas frecuencias.

Este procedimiento se vuelve a realizar pero esta vez para las columnas.

3. Resultados y discusión

En esta primera parte mostraremos algunos

rendimientos en tiempos de ejecución además de algunos resultados que serán determinantes para nuestro proyecto.

Comenzamos con el filtro de piel, La figura 4 muestra el resultado obtenido, pero notamos que esto se optimizaría al entrenarlo con una red neuronal, pues se requiere que el sistema sea automático. De este filtro se extrae la región de interés para trabajar, es decir todos los puntos donde encontraremos piel.

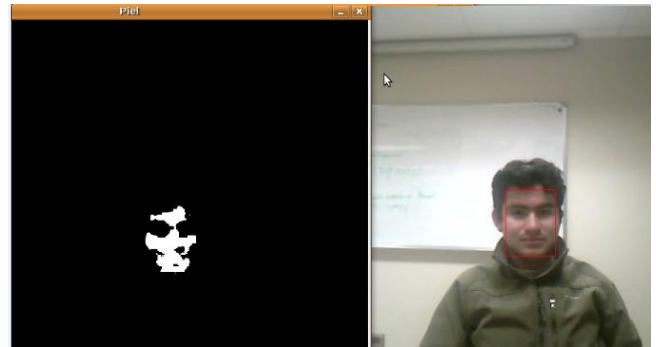


Fig. 4. Filtro de Piel de Colores.

Luego teniendo nuestra región de interés establecida procedemos a determinar donde hay un rostro, debemos hacer una descomposición de wavelets sobre esta región pero sólo sobre el plano de la luminancia (Y) y verificar donde habrían caras con ayuda de la propiedad de las wavelets de conservación de energía [5].

Para esta primera parte modificamos las condiciones del experimento solo mostrando un rostro.

Vemos la Transformada de Daubechies 6 en dos niveles en tiempo real, este es la parte crítica del algoritmo; es decir, donde se consume la mayor parte del tiempo de ejecución por su mayor complejidad.

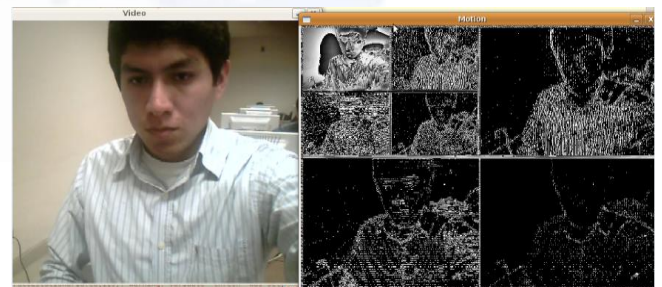


Fig. 5. Transformada de wavelet en tiempo Real.

TABLA I

EJEMPLO DE T. DE WAVELETS-2D DE DAUBECHIES 6

RENDIMIENTOS EN MILLISEGUNDOS PARA EL TAMAÑO DE PÍXELES DADO

| Hardware\Pixels | 320x240 | 640x480 | 1024x768 | 1600x1200 |
|-----------------|---------|---------|----------|-----------|
| GPU (ms) | 16.203 | 67.713 | 172.75 | 428.36 |
| CPU-Matlab(ms) | 46.9 | 171.9 | 375.0 | 890.6 |

Notamos una ventaja considerable el usar la tarjeta de video para procesar esta parte, pues al paralelizar el algoritmo sobre cada pixel se reduce la complejidad temporal, lo cual se aprecia en una reducción del tiempo que va desde la mitad, en los peores casos, hasta la cuarta parte, en los casos más sobresalientes.

4. Conclusiones

La velocidad del procesamiento de los datos usando algoritmos paralelo, es evidentemente más eficiente que sus equivalentes en modo serial para una gran cantidad de algoritmos, por lo que es muy recomendable usarlos en tratamientos de imágenes, debido a su gran carga computacional.

El empleo de la dwt-2d es eficiente para el reconocimiento de caras, pero solo hasta su segundo nivel, pues se aprecia en su tercer nivel, una considerable distorsión de la cara que se desea ubicar.

El filtro de piel usado hasta ahora es operado manualmente, obteniéndose buenos resultados, sin embargo se espera aún mejores resultados empleando una red neuronal que ubique sus parámetros de forma autónoma.

5. Agradecimientos

Al Dr. Javier Solano, por el asesoramiento y recomendaciones dadas.

A CTIC – UNI, por las facilidades brindadas y por el ambiente agradable.

A nuestras familias, por el apoyo incondicional.

6. Referencias

- [1] A. L. Graps, "An Introduction to Wavelets", IEEE Computational Sciences and Engineering, Volume 2, Number 2, summer 1995, p. 50-61.[Online].Disponible:
<http://www.amara.com/IEEEwave/IEEEwavelet.html>
- [2] W. Cova, R. Cavallero, "Sobre Wavelets e Imágenes", Universidad Tecnológica Nacional, Argentina, Agosto 2006, p. 14-35.[Online].Disponible:
http://www.edutecne.utn.edu.ar/DOCUMENTOS/Sobre%20Wavelets%20e%20Imágenes_R1.pdf
- [3] I. Daubechies, "Ten Lectures on Wavelets", Society for Industrial and Applied Mathematics, 1992, ISBN 0898712742
- [4] T. J. Bracamonte, Liz S.R. y P. Huamán, "Aplicaciones de la Red Neuronal de Kohonen al Reconocimientos de Rostros", Universidad Nacional de Trujillo, p. 5-6. [Online]. Disponible:
<http://eventos.spc.org.pe/jpc2007/MyReview/FILES/p12.pdf>
- [5] B. M. Mohammand, S. Kasaei, "An Efficient Wavelet/Neural Network-Based Face Detection Algorithm", Proceeding of the First International Conference on Modeling, Simulation and Applied Optimization, Sharjah, U.A.E. February 1-3, 2005
- [6] S.Gong, S. J. Mckenna, A. Psarrou, "Dinamic Vision from Images to Face REcognition", Imperial College Press,2000, p. 81 – 92
- [7] L. Stan, "Face Detection",A Handbook of Face Recognition, Spinger, 2005, p. 13-32.
- [8] T. Cootes,C. Taylor, H. Kang, V. Petrovic. "Modeling Facial Shape and Appearance",A Handbook of Face Recognition, Spinger, 2005, p. 39-60.
- [9] NVIDIA Cororation, NVIDIA CUDA Compute Unified Device Architecture Programming Guide 2.2.1, May 2009.[online].Disponible:
<http://www.nvidia.com/cude>.
- [10] NVIDIA Cororation, CUDA C Programming Best Practices Guides 2.3, July 2009.[online].Disponible:
http://www.nvidia.com/object/cuda_develop.html.