PREPROCESADOR DE IMÁGENES DE ALTA EFICIENCIA EN EL NIVEL BAJO DE PROCESAMIENTO

BRUNO CERNUSCHI FRÍAS
OSVALDO A. A. BIANCHI
RICARDO A. VEIGA
CARLOS SEIVANE

Facultad de Ingeniería - Universidad de Buenos Aires
Paseo Colón 850 - (1063) Buenos Aires - Argentina

Resumen - Se describe en este trabajo una máquina orientada al procesamiento de imágenes. Este preprocesador está diseñado especialmente para realizar sumatorias de productos eficientemente. La arquitectura está optimizada para tratar las imágenes en una forma flexible, proveyendo una forma económica de implementar algoritmos complejos. También se muestra su aplicación inmediata a la resolución de modelos autorregresivos causales.

I. INTRODUCCIÓN

El modelado es muy importante en la compresión del ancho de banda de imágenes [1], [2].

La idea de eliminar la información redundante, mediante la utilización de un modelo autorregresivo, es una técnica que ha obtenido compresiones que están por debajo de 1 bit/pixel [3], [4].

Los procedimientos empleados en estas técnicas requieren, usualmente, la realización de operaciones de suma y multiplicación en forma masiva. También son necesarias estas operaciones, en gran escala, para la estimación de parámetros de superficies [5]-[10] en los procedimientos de convolución [11], [12], filtrado con ventanas de tamaño fijo, etc..

En todos los casos se deben calcular sumatorias del tipo

$$\sum_{i=1}^{N} Y[f(i)] * Z[g(i)]$$

donde $Y[.]$ y $Z[.]$ pueden ser coeficientes almacenados en alguna memoria, los valores de las posiciones o del brillo de los pixels, etc.; las funciones $f(.)$ y $g(.)$ son las transformaciones genéricas que permiten modificar el índice de barrido "i" según el tipo de técnica que se esté utilizando. Por ejemplo, en el caso de convolución $f(i)$ y $g(i)$ serán iguales a "i" e "$i+j$", respectivamente.

Luego de estas consideraciones, se hace necesario contar con recursos computacionales que realicen estos cálculos en forma eficiente.

En los últimos años se han propuesto arquitecturas de computadoras no convencionales, capaces de manejar gran cantidad de datos simultaneamente [13]-[15]. Generalmente se trata de maquinas basadas en arquitecturas paralelas, que obtienen resultados importantes, pero a un costo elevado. Por ejemplo, el sistema multiprocesador ZMOB [16] ejecuta cerca de 10 Mflops (palabras de 32 bits) con un costo aproximado de U$S 150000.

En este trabajo se presenta una nueva arquitectura de computadora que permite concretar esos objetivos de una forma mucho más económica.

En estos momentos estamos trabajando activamente en el tema de compresión del ancho de banda de imágenes. Aunque existen muchas técnicas [17], aquí mostraremos la utilidad de esta arquitectura en la resolución de los modelos autorregresivos causales (Predicción Lineal).

II. MODELOS AUTORREGRESIVOS PARA UNA IMAGEN DE INTENSIDAD

Denominaremos "modelo 1" al más comunmente utilizado en predicción lineal [18], y que es el modelo autorregresivo para una señal con valor medio nulo, como el que se muestra en la figura 1.

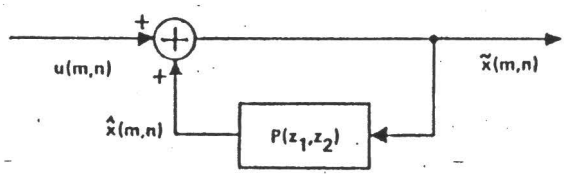


Figura 1. *"Modelo 1" que supone a la imagen con valor medio nulo.*

El predictor que esta en el lazo de realimentación, que tiene en cuenta la naturaleza autorregresiva del modelo considerado, tiene una función de transferencia

$$P(z_1, z_2) \triangleq \sum_{\substack{K\,I \\ (K,I) \in \mathcal{H}}} \sum a(K,I)\, z_1^{-k} z_2^{-l}$$

donde $a(K,I)$ representa a un conjunto de coeficientes de predicción en 2-D, $\mathcal{H}$ es un conjunto de pares enteros que representa la zona del espacio que se considera para realizar la predicción y que será definido más adelante. De aquí en más, cada vez que se exprese una sumatoria en "KI" será equivalente a $(K,I) \in \mathcal{H}$, salvo indicación expresa.

Consideremos una secuencia de muestras de intensidades de una imagen $x(m,n)$, sintetizadas por el modelo 1. La ecuación en diferencias que caracteriza a este modelo es

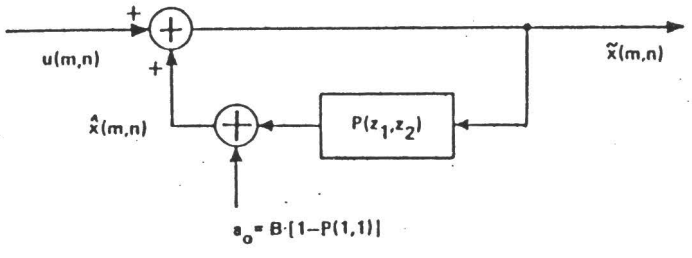
$$\tilde{x}(m,n) = \sum_{K\,I} \sum a(K,I)\, \tilde{x}(m-K, n-I) + u(m,n)$$

$$= \hat{x}(m,n) + u(m,n)$$

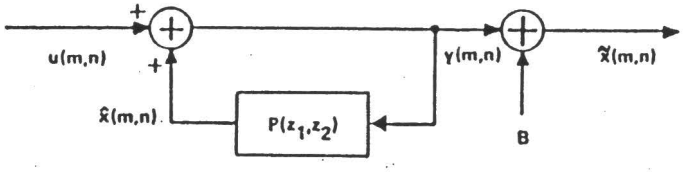
Este modelo requiere que la señal posea valor medio nulo, pero las imágenes no cumplen con este requisito, ya que siempre toman valores positivos. Por lo tanto, es necesario estimar la "función valor medio", que es variante en el espacio, mediante un filtrado pasa bajos [19].

Otros modelos que analizaremos aquí, tienen en cuenta el hecho de que las imágenes no tienen valor medio nulo. Un modelo posible es el que se muestra en la figura 2(a), y que denominaremos "modelo 2a", donde a_0 representa el "valor medio" o "polarización" local que se suma al sistema realimentado. Un modelo equivalente a éste, que se muestra en la figura 2(b) y que denominaremos "modelo 2b", puede obtenerse suponiendo la suma de una secuencia autorregresiva de valor medio nulo $y(m,n)$ y una constante

B que representa el "valor medio" o "polarización" local, pero ahora a la salida del modelo.



(a)



(b)

Figura 2. *(a) "Modelo 2a" con polarización a la entrada. (b) "Modelo 2b" con polarización a la salida.*

Para el "modelo 2a", la ecuación en diferencias, que relaciona la salida $\tilde{x}(m,n)$ con la excitación de entrada $u(m,n)$, es

$$\tilde{x}(m,n) = \sum_{K}\sum_{l} a(K,l)\,\tilde{x}(m-K,n-l) + a_0 + u(m,n) \qquad (1)$$

$$= \hat{x}(m,n) + u(m,n)$$

La ecuación en diferencias resultante del "modelo 2b" es

$$\tilde{x}(m,n) = y(m,n) + B = \hat{x}(m,n) + u(m,n) \qquad (2)$$

$$= \sum_{K}\sum_{l} a(K,l)\,y(m-K,n-l) + B + u(m,n)$$

Comparando las ecuaciones (1) y (2) se obtiene la relación entre a_0 y B.

$$a_0 = B\,[\,1 - \sum_{K}\sum_{l} a(K,l)\,] = B\,[\,1 - P(1,1)\,]$$

A. Zona de predicción

El conjunto Π de pares enteros que definen la "zona de predicción" determina la causalidad espacial del modelo, ya que ésta no es

inherente a la formación de la imagen. Generalmente se desea que las ecuaciones en diferencias, que representan a estos modelos, sean computables recursivamente. Por lo tanto, esto limita la zona de predicción a algún subconjunto del semiplano asimétrico

$$\hat{\hat{\Pi}} = \{ (K,l) : (K>0,\ l\geq 0) \vee (K\leq 0,\ l\geq 1) \}$$

Un subespacio de este semiplano es

$$\hat{\Pi} = \{ (K,l) : (0<K\leq K_M,\ l=0) \vee (-K_M\leq K\leq K_M,\ 1\leq l\leq l_M) \}$$

Otro subespacio que puede considerarse, y que es muy utilizado, es la mitad del semiplano anterior, o sea, un cuarto de plano que llamaremos cuadrante. Sacrificando alguna generalidad, puede limitarse dicho cuadrante a los elementos del conjunto

$$\Pi = \{ (K,l) : 0\leq K\leq Q-1 \wedge 0\leq l\leq Q-1 \wedge (K,l) \neq (0,0) \} \qquad (3)$$

El ordenamiento de los coeficientes de predicción no es única, pero algunas elecciones pueden ser más apropiadas que otras. Dos de los ordenamientos posibles son:
- Ordenamiento lexicográfico inverso ("inverse lexicografic ordering").
- Ordenamiento por el vecino más cercano ("close-neighbor ordering").

B. Predicción lineal de imágenes de intensidad

B.1. Método "Least-Mean Square" (LMS)

En la figura 3 se muestra un sistema ADPCM con predictor adaptativo .

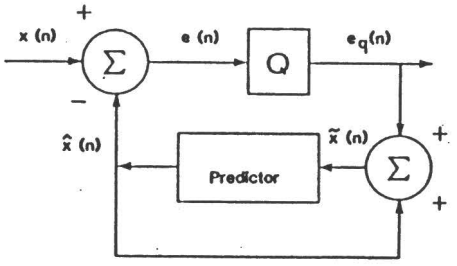


Figura 3. *Sistema ADPCM con predictor adaptativo.*

Una forma de expresar el valor estimado de $x(m,n)$ es

$$\hat{x}(n) = \underline{w}^T(n) \cdot \underline{\tilde{x}}(n) = \sum_{i=1}^{N} w_i(n)\ \tilde{x}(n-\hat{c}_i)$$

donde los $w_i(n)$ son los N coeficientes adaptativos del predictor, $\tilde{x}(m,n)$ son los valores reconstruidos de las muestras de la imagen, y los $\hat{c}_i$ son los retardos con los cuales se seleccionan los pixels que sirven de base para la predicción. En cada pixel se calcula el error de predicción

$$e(n) \triangleq x(n) - \hat{x}(n)$$

Este error es cuantizado, obteniéndose $e_q(n)$, y luego enviado al receptor. Pero también es utilizado para actualizar los valores de los coeficientes del predictor para la próxima iteración

$$w_i(n+1) = w_i(n) + \gamma' e_q(n) \tilde{x}(n-\zeta_i) \qquad 1 \leq i \leq N$$

donde γ' es un parámetro que controla la convergencia y el estado de régimen del predictor. Este método es una aproximación al método del gradiente.

B.2. Método de los cuadrados mínimos (Least Squares (LS))

Los parámetros del modelo de la imagen pueden ser estimados con el criterio de minimizar el valor cuadrático medio de la secuencia de error de predicción

$$e(m,n) \triangleq x(m,n) - \hat{x}(m,n)$$

donde $e(m,n)$ es el error de predicción, $x(m,n)$ es una muestra de la imagen y $\hat{x}(m,n)$ es el correspondiente valor estimado de $x(m,n)$.

El error de predicción cuadrático medio residual se define como

$$E \triangleq \sum_{mn}\sum e^2(m,n) \qquad (m,n) \in \Omega$$

donde Ω es la región de soporte de la secuencia $e(m,n)$ donde se minimiza el error E, llamada "zona de análisis" y que definiremos más adelante.

En lo sucesivo, para abreviar la notación, las sumas sobre 'm' y 'n' significarán que son sobre $(m,n) \in \Omega$.

Los coeficientes óptimos del modelo (según el criterio adoptado) son aquellos que minimizan E. En consecuencia,

$$\partial E / \partial a(i,j) = 0 \qquad \forall (i,j) \in \overline{J1} + (0,0)$$

C. Zona de análisis

La "zona de análisis" es la región de soporte de la secuencia $e(m,n)$ donde se minimiza el error E.

Para cada zona de análisis se suponen fijos los coeficientes de predicción del modelo.

Es interesante hacer notar que la definición de la zona de análisis no es única, pero algunas elecciones pueden ser más apropiadas que otras. Por ejemplo, se puede definir una zona de análisis cuadrada de MxM pixels

$$\{ L \leq m \leq U, \quad L \leq n \leq U \} \Rightarrow M \triangleq U-L \Rightarrow Ns = M^2$$

donde Ns es el número total de pixels en la zona de análisis.

D. Cálculo de los coeficientes del predictor

a) Para el modelo 1.

Para este modelo asumiremos como zona de predicción, por ejemplo, la de $\hat{\Pi}$. Entonces, minimizando el error E (admitiendo que el error de cuantización es despreciable), queda

$$\sum_{kl} a(k,l)\ \phi(k,l:i,j) = \phi(0,0:i,j) \qquad \forall\ (i,j),(k,l) \in \hat{\Pi}$$

con

$$\phi(k,l:i,j) \triangleq \sum_{mn} x(m-k,n-l)\ x(m-i,n-j) \qquad \forall\ k,l,i,j$$

b) Para el modelo 2a

Para este modelo asumiremos como zona de predicción, por ejemplo, la de $\hat{\Pi}$. Siguiendo un procedimiento similar al que se utilizó con el modelo 1, se obtiene

$$\phi(0,0:i,j) = \sum_{kl} a(k,l)\cdot\phi(k,l:i,j) + a_0 S(i,j) \qquad \forall (i,j) \in \hat{\Pi} \quad (4)$$

$$S(0,0) = \sum_{kl} a(k,l)\cdot S(k,l) + Ns\cdot a_0 \qquad (5)$$

donde

$$\phi(k,l:i,j) \triangleq \sum_{mn} x(m-k,n-l)\ x(m-i,n-j) \qquad \forall\ k,l,i,j$$

$$S(i,j) \triangleq \sum_{mn} x(m-i,n-j)$$

y Ns es el número de muestras en la "zona de análisis", que es la región de soporte de la secuencia $e(m,n)$ donde se minimiza el error E.

Utilizando el cuadrante $\hat{\Pi}$, definido en (3), y el siguiente ordenamiento lexicográfico inverso,

$$T(k,l) = q = 1\cdot Q + K \qquad (k,l) \in \hat{\Pi} \cup (0,0)$$

$$(k,l) = T^{-1}(q) \qquad 0 \leq q \leq P, \qquad P = Q^2 - 1$$

$$\left.\begin{array}{l} T(k,l) = q \\ T(i,j) = r \end{array}\right\} \qquad \phi(k,l:i,j) = \phi(q:r) \qquad S(k,l) = S(q)$$

$$\Theta_1 = \{\, q \in \mathbb{Z} : 1 \leq q \leq P \,\} \qquad \Theta_2 = \{\, r \in \mathbb{Z} : 0 \leq r \leq P \,\}$$

se puede escribir el conjunto de ecuaciones normales (4) y (5) en la forma

$$\sum_{q} a(q)\ \phi(q:r) + a_0 S(r) = \phi(0:r), \qquad \forall\ q \in \Theta_1,\ r \in \Theta_1$$

$$\sum_{q} a(q)\ S(q) + a_0 Ns = S(0), \qquad r=0$$

y en forma matricial

$$C\,\underline{a} = \underline{c}\ .$$

$$\left[\begin{array}{cccc|c} \phi_{11} & \phi_{21} & \cdots & \phi_{P1} & S_1 \\ \phi_{12} & \phi_{22} & & & \\ \cdot & & & \cdot & \cdot \\ \cdot & & & \cdot & \cdot \\ \cdot & & & \cdot & \cdot \\ \phi_{1P} & & \cdots & \phi_{PP} & S_P \\ \hline S_1 & & \cdots & S_P & N_S \end{array}\right] \left[\begin{array}{c} a_1 \\ \\ \cdot \\ \cdot \\ \cdot \\ a_P \\ \hline a_0 \end{array}\right] = \left[\begin{array}{c} \phi_{01} \\ \\ \cdot \\ \cdot \\ \cdot \\ \phi_{0P} \\ \hline S_0 \end{array}\right]$$

donde

$$\underline{a} = [\underline{\alpha}^T, a_0] \qquad y \qquad \underline{c} = [\underline{r}^T, S_0]$$

La submatriz de PxP elementos, ubicada arriba a la izquierda, se llama matriz de correlación, y la denotaremos con R. A los vectores $\underline{r}$ y $\underline{\alpha}$ los llamaremos vector de correlación y vector de los coeficientes de predicción, respectivamente.

Para la obtención de los coeficientes del predictor, se puede invertir la matriz de correlación de PxP elementos. Un algoritmo sencillo requiere una cantidad de operaciones $O(P^3)$. Sin embargo, se han propuesto soluciones más eficientes mediante procedimientos recursivos, con variantes del tipo "lattice" [20], [21]. Pero, en realidad, el cuello de botella se encuentra en el cálculo de los elementos de dicha matriz. Esto es así debido a que son necesarias O(NsP) operaciones; donde Ns es la cantidad total de pixels en la zona de análisis, y P el orden del predictor. Ns es mayor que P, y generalmente mucho mayor que P^2; ya que se ha comprobado que no se logran mejoras sustanciales para valores de P superiores a 9 (Q=3), y Ns=MxM suele ser superior a 32x32, e incluso llega a 256x256 para una zona de análisis de imagen completa. Pero aún cuando la cantidad de operaciones fuese del mismo orden, los elementos deben ser calculados igualmente.

En consecuencia, aquí propondremos una arquitectura capaz de resolver, en forma económica, esta clase de problemas donde se deben efectuar cálculos masivos.

III. PREPROCESADOR PROSUM I

El preprocesador PROSUM I es una máquina orientada al procesamiento de imágenes, siendo capaz de operar en dos modos: a) tiempo real y b) cuasi tiempo real. El preprocesador posee una arquitectura modular; implementado principalmente con bloques VLSI. La arquitectura está optimizada para tratar las imágenes en una forma flexible, proveyendo una forma económica de implementar algoritmos complejos.

La figura 4 muestra un diagrama en bloques de la máquina. Ésta opera a 75 instrucciones por segundo, donde cada instrucción se aplica a una imagen entera (SIMD).

La computadora central es una Micro Vax II operando con ambiente Ultrix-C.

Las imágenes están almacenadas en registros especiales llamados Frame Registers; además, la interfase de entrada/salida también está basada en registros. Este concepto lleva a un tratamiento rápido y homogéneo de las imágenes.

Cada Frame Register es una matriz de almacenamiento de 256 X 256 X 32 bits, capaz de operar con punto flotante. Estos están implementados con memoria estática de alta velocidad, con tiempo de acceso menor que 100ns.

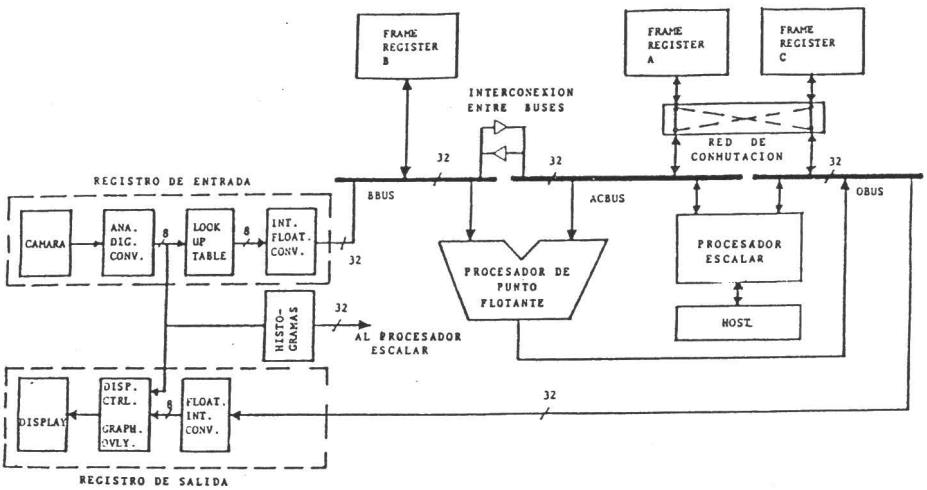


Figura 4. *Diagrama en bloques de la arquitectura propuesta.*

Los Frame Registers tienen la capacidad de definir porciones limitadas de la imagen, llamadas ventanas. Así, esta operación crea un nuevo tipo de datos, llamado registro reducido. De esta manera, esta idea permite concentrar la potencia de procesamiento sobre zonas de interés especial. Su implementación se realiza mediante dos conjuntos de comparadores y registros que almacenan el largo y el ancho de la ventana.

Los Frame Registers poseen punteros que permiten desplazar el punto de referencia de imagen por X pixels y N líneas. La dirección efectiva del pixel considerado se obtiene mediante la suma, con hardware de alta velocidad, de la dirección base y el contenido del puntero.

El Procesador de Punto Flotante es un circuito integrado de Weitek Corporation, que es capaz de realizar una operación de multiplicación y suma cada 100 ns. Así, utilizando el formato IEEE, se puede obtener una capacidad de 20 Mflops.

La máquina posee un Data Path de 32 bits que interconecta los Frame Registers y llevan información al Procesador de Punto Flotante.

Refiriéndonos al diagrama en bloques, se puede advertir que existen tres Buses principales, de 32 líneas cada uno: BBUS, ACBUS, OBUS.

Debe observarse que no existen líneas de dirección, puesto que éstas son generadas internamente en los Frame Registers.

Las interconexiones de los buses se resumen en la tabla I.

Existen dos niveles de operación:

a) Nivel rápido (20 Mflops) - operación a nivel de pixel -
b) Nivel lento (operación a 75 imágenes completas por segundo)

Por lo tanto, la unidad de control está dividida en dos secciones:

1) Unidad de control principal.
2) Microsecuenciador.

La unidad de control principal, que está construida con un

Conexionado de Buses

BBUS

B ------> IA (PROCESADOR DE PUNTO FLOTANTE)
ENTRADA IMÁGENES ------> B
ENTRADA IMÁGENES ------> IA (PROCESADOR DE PUNTO FLOTANTE)

ACBUS *

A ------> IB (PROCESADOR DE PUNTO FLOTANTE)
A <------> PROCESADOR ESCALAR O COMPUTADOR CENTRAL

OBUS *

OZ (PROCESADOR DE PUNTO FLOTANTE) -------> C
C -------> DISPLAY DE IMÁGENES
OZ (PROCESADOR DE PUNTO FLOTANTE) -------> DISPLAY DE IMÁGENES
PROCESADOR ESCALAR O COMPUTADOR CENTRAL <-------> C
PROCESADOR ESCALAR O COMPUTADOR CENTRAL <-------> DISPLAY DE IMÁGENES

* Existe una red de conmutación que intercambia A con C.

Tabla I

microprocesador, recibe las instrucciones de la computadora central. Estas pueden ser de tres tipos :

a)Instrucciones de imagen completa.
b)Instrucciones para el procesador escalar.
c)Instrucciones de soporte, comunicación y control.

Cuando la unidad de control principal recibe una instrucción de imagen completa, carga el Puntero del Frame Register, marca las ventanas, establece los Data Path, carga el registro de instrucciones, y luego envía una señal de comienzo al microsecuenciador.

Cuando recibe instrucciones para el procesador escalar, la unidad de control conecta los Data Path para que este pueda operar sobre el Frame Register seleccionado.

Las instrucciones de comunicación controlan una interfase de DMA con la computadora central.

El microsecuenciador implementa funciones simples que necesitan alta velocidad. Las microinstrucciones las recibe de la unidad de control principal, las cuales manejan las excepciones, y controlan tanto la generación de las direcciones como al procesador de punto flotante. Después de procesar una imagen o recibir una excepción, carga un registro de estado que informa a la unidad de control principal. Este microsecuenciador es de tipo cableado configurado con lógica MSI.

La técnica de utilizar registros como entrada/salida consiste en considerarlos como Frame Registers especializados, que tratan la entrada (Video source) como registro de lectura solamente, y la salida (Display) como registro de escritura solamente. Esto permite un tratamiento homogéneo y un conjunto de instrucciones más regular.

En esta máquina se incluyen: un Generador Automático de Histogramas, una Lookup Table de entrada y otra de salida. Estos tres elementos están orientados al procesamiento en tiempo real de cada pixel. El generador automático de histogramas examina un pixel por vez y cuenta las apariciones de cada intensidad. Esta información es útil para el mejoramiento y la clasificación de las imágenes. Las Lookup table de entrada y de salida modifican el valor del pixel a medida que pasan por ellas, y permiten una transformación lineal o no lineal de las intensidades.

Aunque el preprocesador de imágenes opera en blanco y negro, el sistema de Display puede operar con colores. Esto permite mostrar los diferentes niveles de intensidad en distintas tonalidades. Además, el sistema de Display posee capacidad de superponer tanto textos como gráficos sobre la imagen, sin alterarla. Este sistema provee una excelente interface hombre-máquina.

Como se mencionó anteriormente, existen tres tipos de instrucciones. El conjunto de instrucciones de imagen completa es regular, pues todas las instrucciones consisten en una operación con uno o dos registros como fuente y uno como destino. La fuente y el destino pueden ser la cámara y el display, respectivamente. Las operaciones de imagen completa se implementan mediante el procesador de punto flotante y las Lookup Tables.

Las instrucciones para el procesador escalar operan sobre diferentes tipos de datos; actualmente estamos definiéndolas y regularizándolas.

Las instrucciones de soporte, comunicaciones y control son altamente especializadas y controlan el funcionamiento general del sistema.

IV. Conclusiones

En este trabajo se describió la arquitectura de un preprocesador, diseñado especialmente para efectuar cálculos masivos en el nivel bajo de procesamiento de imágenes. Su construcción es económica y permite resolver, eficientemente, problemas que generalmente se reservan para máquinas mucho más grandes, complejas y caras.

También se mostró como puede aplicarse inmediatamente para la estimación de parámetros en un modelo autorregresivo causal.

La arquitectura aquí propuesta es importante para resolver esta clase de problemas donde se deben hacer calculos masivos.

Referencias

[1] A. K. Jain, "Image data compression: A review," *Proceedings of the IEEE*, vol. 69, no. 3, pp. 349-389, Marzo 1981.

[2] A. K. Jain, "Advances in mathematical models for image processing," *Proceedings of the IEEE*, vol. 69, no. 5, pp. 502-528, Mayo 1981.

[3] P. A. Maragos, R. W. Schafer, and R. M. Mersereau, "Two-Dimensional linear prediction and its application to adaptive predictive coding of images," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-32, no. 6, pp. 1213-1229, Diciembre 1984.

[4] S. T. Alexander and S. A. Rajala, "Image compression results using the LMS adaptive algorithm," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-33, no. 3, pp. 712-714, Junio 1985.

[5] B. Cernuschi-Frías, "Orientation, location and parameter estimation of quadric surfaces in 3-D space from a sequence of images," Ph. D. Thesis, UMI Press, also Brown University Tech. Report #LEMS-5, Mayo 1984.

[6] R. M. Bolle and D. B. Cooper, "Bayesian recognition of local 3-D shape by approximating image intensity functions with quadric polynomials," *IEEE Trans. on PAMI*, vol. PAMI-6, no. 4, pp. 418-429, Julio 1984.

[7] B. Cernuschi-Frías and D. B. Cooper, "3-D space location and parameter estimation of lambertian spheres and cylinders from a single 2-D image by fitting lines and ellipses to thresholded data," *IEEE Trans. on PAMI*, vol. PAMI-6, no. 4, pp. 430-441, Julio 1984.

[8] B. Cernuschi-Frías and D. B. Cooper, "Classification and ranging of quadric surfaces from TV images obtained by a moving camera," *Proc. of the 1985 IEEE Conference on Systems, Man and Cybernetics*, Tucson, Arizona, pp. 943-947, Noviembre 1985.

[9] B. Cernuschi-Frías, P. N. Belhumeur and D. B. Cooper, "3-D object position estimation and recognition based on parametrized surfaces and multiples views," *Proc. of the 1986 IEEE International Conference on Robotics and Automation*, San Francisco, California, pp. 639-644, Abril 1986.

[10] B. Cernuschi-Frías and D. B. Cooper, "Lambertian sphere parameter estimation from a single 2-D image," *Proc. of the 1986 IEEE International Conference on Systems, Man and Cybernetics*, Atlanta, Georgia, pp. 983-986, Octubre 1986.

[11] H. K. Nagpal, G. A. Jullien, and W. C. Miller, "Processor architectures for two-dimensional convolvers using a single multiplexed computational element with finite field arithmetic," *IEEE Trans. on Computers*, vol. C-32, no. 11, pp. 989-1000, Noviembre 1983.

[12] K. Doshi and P. Varman, "A modular systolic architecture for image convolutions," *Proceedings of the 14th International Symposium on Computer Architecture*, pp. 56-63, ACM/IEEE, Junio 1987.

[13] L. Uhr, "Comparing serial computers, arrays, and networks using measures of 'Active Resourses'," *IEEE Trans. on Computers*, vol. C-31, no. 10, pp. 1022-1025, Octubre 1982.

[14] A. V. Kulkarni and D. W. L. Yen, "Systolic processing and implementation for signal and image processing," *IEEE Trans. on Computers*, vol. C-31, no. 10, pp. 1000-1009, Octubre 1982.

[15] A. Levinthal, P. Hanrahan, M. Paquette, and J. Lawson, "Parallel computers for graphics applications," *Proceedings of the 2nd International Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 193-198, ACM/IEEE, Octubre 1987.

[16] T. Kushner, A. Y. Wu, and A. Rosenfeld, "Image processing on ZMOB", *IEEE Trans. on Computers*, vol. C-31, no. 10, pp. 943-951, Octubre 1982.

[17] J. Guichard and D. Nasse, "Digital image coding," *l'Echo des Recherches*, English issue, pp. 9-24, 1987.

[18] J. Makhoul, "Linear Prediction: A Tutorial Review," *Proceedings of the IEEE*, vol. 63, no.4, pp. 561-580, Abril 1975.

[19] L. F. Chaparro and M. Boudaoud, "Two-Dimensional linear prediction covariance method and its recursive solution," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. SMC-17, no. 4, pp. 617-621, Julio/Agosto 1987.

[20] B. Friedlander, "Lattice filters for adaptive processing," *Proceedings of the IEEE*, vol. 70, no. 8, pp. 829-867, Agosto 1982.

[21] B. Cernuschi-Frías and R. A. Veiga, "An alternative realization of the least squares lattice algorithm," *Proc. of EUSIPCO 88*, Grenoble, France, 1988.