

ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERIA ELECTRICA

"ESTUDIO E IMPLEMENTACION DE UN DIGITALIZADOR DE VIDEO"

TESIS PREVIA A LA OBTENCION DEL TITULO DE INGENIERO EN
ELECTRONICA Y TELECOMUNICACIONES

ALEX ATAHUALPA MAYORGA LEON

QUITO - ECUADOR

1.992 1.993

DEDICATORIA

A mi Hijo Alex Daniel a mis
Padres y a mi tía Evita León,
con mucho cariño.

AGRADECIMIENTO

A mis padres y a todos los profesores que han colaborado en este trabajo como la Ing. Tania Perez, y especialmente a mi Director de tesis.

CERTIFICACION

Certifico que la presente tesis ha sido desarrollada por el Sr. Alex Atahualpa Mayorga León, bajo mi dirección.

Ing. Ramiro Morejón.

INDICE

I CAPITULO 1

Pgs.

1.1	Fundamentos para la formación de una señal de Video	1
1.2	El CCD, Analógico o Digital? Principio y captación de imagen con esta tecnología	7
1.2.1	Teoría CCD	7
1.2.2	Conversión fotoeléctrica (se genera una carga eléctrica usando luz)	11
1.2.3	Almacenaje de carga	11
1.2.4	Operación de transferencia de carga (se transfiere una carga eléctrica)	12
1.2.5	Operación de lectura del sensor mediante señal impulsadora de tres estados	14
1.2.6	Impulsor Vertical	16
1.2.7	Transferencia de carga del registro vertical al registro horizontal	17
1.2.8	Mecanismo de transferencia del registro horizontal a la porción de salida	18
1.3	Criterios para el tratamiento de la señal de luminancia	22
1.3.1	Corrección de gamma	23
1.3.2	Fijador de nivel de negro	24
1.4	Criterios para el tratamiento de crominancia	27

1.4.1	Elección de la frecuencia subportadora de crominancia	27
1.5	Tratamiento de sincronismos	39
1.5.1	Intervalo horizontal	39
1.5.2	Intervalo vertical	41
1.5.2.1	Señal de prueba en el intervalo vertical(VITS)	42
1.5.3	Burst	43
1.5.3.1	Señal de sincronismos en NTSC	43
1.5.3.2	Técnicas de black burst	44
1.5.3.3	Burst timing (regulación de tiempo para el burst)	45

CAPITULO II

2.	Fundamentos teóricos para la digitalización de una señal de video	46
2.1	Conversión análogo digital	46
2.2	Codificación de una señal digital	57
2.2.1	Comprensión de imagenes	58
2.2.1.1	IFS (Iterated function systems)	59
2.2.1.2	Codificación por segmentación	63
2.2.1.3	Codificación por descomposición direccional	73
2.2.2.1	Codificación de imagenes usando vectores de cuantización	82
2.2.3	Codificación de imagenes basados en transformación ortogonales traslapadas	94

2.2.4	Otros tipos de codificación	100
2.2.4.1	Piramide Laplaciana como un código compacto de imagen	100
2.3	Administración de memoria para un video codificado	106
2.3.1	Compensación de color en codificación de imágenes	118
2.3.2	Predicción de imágenes	123

CAPITULO III

3.	Implementación de un prototipo de una digitalizador para señal de video	128
3.1	Diagrama de bloques	129
3.2	Diseño y justificación del circuito	129
3.2.1	Codificador	130
3.2.2	Filtros	130
3.2.3	Matriz	131
3.3	Lógica de operación del circuito implementado	133

CAPITULO IV

4.	Análisis de resultados y conclusiones	150
ANEXOS		156

INTRODUCCION

El tema que se ha escogido para el siguiente trabajo se ha basado en la gran difusión del video, su necesidad comercial como científica, lo que le ha llevado a encontrar métodos y formas para su proceso. Por ello el conocimiento de la existencia de algunas de ellas, ayudarán a enrumbar cualquier trabajo posterior.

El capítulo I se dedica a dar los fundamentos del video y la televisión, en su presentación visual transmisión y recepción en el aspecto de características como señal.

El capítulo II después de tener esta idea, indicará algunos métodos que se han desarrollado, tal que sean compatibles con la señal convencional. Estos métodos se los topará en forma descriptiva y como incluyendo los fundamentos en los cuales se basan cada tipo de proceso.

El capítulo III es la explicación de la implementación de un prototipo de digitalizador de video, puesto que es el primer paso forzoso para tener en forma tal que se pueda simular procesos digitales.

C A P I T U L O I

1.1.- Fundamentos para la formación de una señal de televisión.-

Para poder entender claramente la formación de una imagen tenemos que comprender la operación de la exploración entrelazada que se utiliza en el sistema NTSC que es el sistema utilizado en nuestro país, este sistema tiene 525 líneas, valor escogido en base a ciertos compromisos los que se describen seguidamente:

Para $n=525$ líneas exploradas en la pantalla, se tiene facilidad en la obtención de convertidores de frecuencia, entre otras ventajas de transmisión. En este sistema se define a:

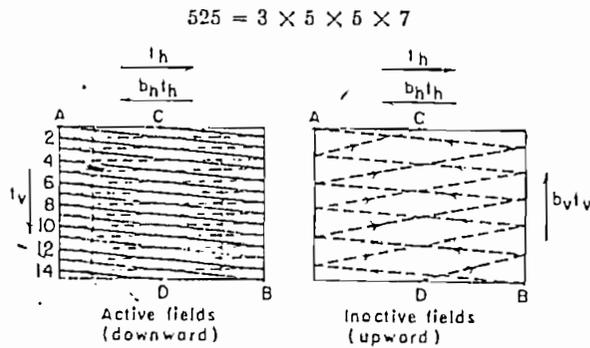
n_a = líneas activas o vistas en la pantalla
siendo exactamente dado por la ecuación

$$n_a = n / (1 + b_v)$$

donde:

b_v es la relación de retorno vertical.

El gráfico ilustra la operación del sistema NTSC y el número de líneas efectivas:



La "exploración entrelazada" se realiza de la siguiente manera: Se exploran las líneas pares de una imagen (llamado campo par) y luego las líneas impares de la misma imagen (campo impar).

La pantalla es explorada horizontalmente desde

izquierda a derecha durante un intervalo t_h , el haz exploración retorna al lado izquierdo de la pantalla en una fracción b_h , cuyo valor b_h puede ser entre 0.10 y 0.25 de t_h (seg); el FCC recomienda un límite de 0.15. Este barrido horizontal es acompañado por t_v , tiempo de movimiento vertical de una línea. Para la relación de retorno b_v puede tomar valores entre los tiempos 0.05 y 0.10 de t_v (seg), el FCC recomienda un límite de 0.08. (referencia: Radio Engineering Handbook, Henney).

Este número de líneas exploradas (525) será el mismo para sistemas de color y en blanco/negro (B/W).

Para garantizar la total compatibilidad en las transmisiones de blanco/negro y color. Se deben compartir características:

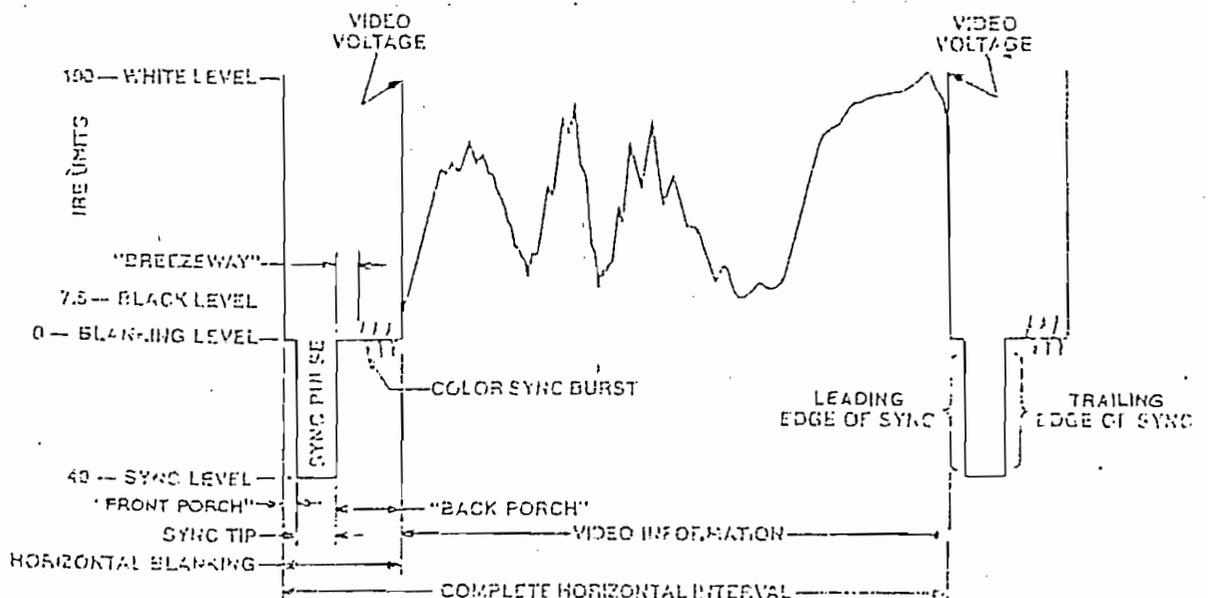
- a) La señal de televisión en color debe producir una imagen en blanco/negro en el receptor B/W.
- b) Un receptor en color debe ser capaz de producir una imagen en blanco/negro cuando recibe una señal B/W normal. Esto es lo que se llama también "compatibilidad inversa".

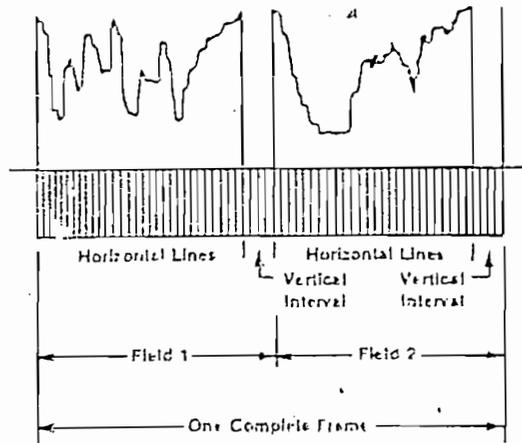
Para cumplir con estas características deben tener las siguientes condiciones:

- a) Utilizar el mismo ancho de banda de T.V. B/W.
- b) Emplear la misma frecuencia de exploración.
- c) Emplear frecuencias para la misma separación entre las portadoras de Audio y Video.

- d) Emplear las mismas señales de sincronización para línea y campo .
- e) La señal de color para un mismo cuadro debe tener la información de brillo que una señal B/W.
- f) Contener la información adicional de color conjuntamente con las señales auxiliares necesarias para permitir la decodificación de dicha información.
- g) Transportar la información de color de tal manera que no deteriore la imagen del receptor B/W.

La televisión en blanco/negro que contiene la información de video en la señal de luminancia (Y); Para televisión de color en base a las señales de rojo, verde y azul, (R,G,B); se puede reconstruir la señal Y en los receptores y monitores a color. El gráfico muestra las partes constitutivas de una línea de video:





La información contenida en una señal de video blanco /negro o color son:

- a) Impulsos correspondientes a la brillantes del elemento captadas por la cámara.
- b) El Blanqueo (Blanking) del haz explorador en la recepción durante el movimiento de retorno, por el nivel de blanking o pedestal.
- c) La sincronización del haz explorador, tanto por el sincronismo vertical y señales de sincronización horizontal.
- d) La señal de color que se adhiere a la señal de luminancia transmitida, que es producida por la captación de la información cuantitativa de los tres colores básicos R-G-B, para luego ser procesados y transmitidos en dos componentes en cuadratura.

e) El audio es transmitido en una portadora 4.5 MHz superior a la portadora de video.

Un importante aspecto, el sincronismo que puede ser generado en el mismo receptor o en el transmisor (cámaras inclusive). En el receptor la señal de sincronismo toma como referencia el sincronismo recibido y se lo procesa para mejorarlo a este, es decir flancos mejor definidos, evitar "jitter" (desplazamientos de fase) y deformaciones en su formas de onda, o generar su propio sincronismo para sí o todo el sistema.

La información de video es captada en la cámara por los tubos de vacío o actualmente por circuitos integrados monolíticos, son pasadas a procesadores para sincronizarse al sistema y generar efectos de video.

1.2.- El CCD. Analógico o Digital. Principio y captación de imagen con esta tecnología.

Actualmente la forma de captar una imagen tiene en esencia el fundamento del tubo de vacío para captación de imagen, es decir, el nivel del electrones desprendidos y absorbidos por la película fotosensible que es proporcional a la intensidad de luz que incide en este.

La diferencia radica en que en el tubo los electrones son producidos por un rayo electrónico que explora toda la pantalla fotosensible o fotoconductiva, y los electrones serán tomados o neutralizados de este haz, para luego ser

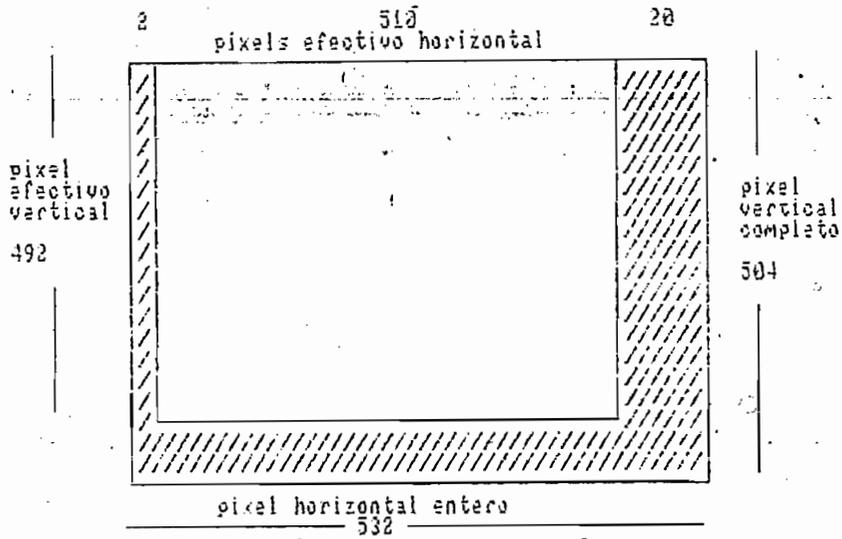
transformados a voltajes y procesados por la cámara de televisión; en el CCD estos electrones serán producidos por efecto fotoeléctrico sobre una película fotosensible y por ende manteniéndose la proporcionalidad de luz con cantidad de electrones.

A continuación se detalla el principio de funcionamiento del dispositivo y sus sincronismos para la exploración de todo el cuadro de imagen que será captada, estos deberán ser compatibles con la teoría en los tubos y sistemas de T.V. ya instalados. Es decir el formato con el cual capta la imagen debe ser por líneas y/o campos.

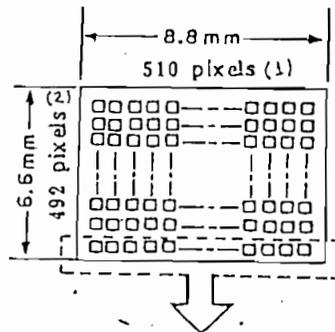
1.2.1.- Teoría CCD.-

Un CCD (Charge Coupling Device) es un dispositivo semiconductor de almacenamiento de cargas que contienen "condensadores" (semiconductores que almacenan carga en pozos de potencial) dispuestos en una forma regular. Procesa las cargas eléctricas de acuerdo a las siguientes funciones:

- 1) Conversión fotoeléctrica (se genera una carga eléctrica usando luz).
- 2) Almacenaje de la carga (se almacena una carga eléctrica).
- 3) Operación de transferencia/tiempo (se transfiere una carga eléctrica).

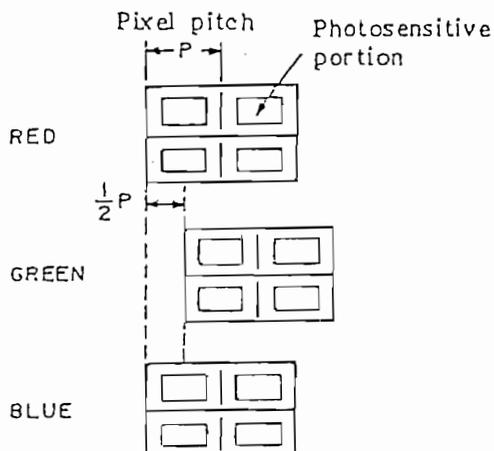


Un Pixel es una celda o porción semiconductor que capta la imagen por efecto fotoeléctrico; y un CCD es el arreglo bidimensional de estos.



Esta variación de pixeles de totales a efectivos está dada entre otras razones por astigmatismo y aberraciones ópticas.

Cuando un sensor de imagen de estado sólido tal como un CCD se usa en una cámara a color, se requiere una gran cantidad de pixels para obtener una imagen de gran calidad. Como el número efectivo de pixels está limitado a la integración y tamaño de cada circuito integrado, se ha conseguido densidades de 504 x 532 (efectivos tan solo son 492x510) o de 790 x 768 (efectivos son 736x565) que influirán directamente en la calidad de imagen que será captada. Los pixels de la periferia se pierden por el sincronismo que maneja la transferencia y operación del CCD, aceptando tan solo el número de pixels efectivos, y desechando los otros, además son para preveer errores en la fabricación de los semiconductores, es decir, pixels con fallas. Los pixels faltantes en el sentido horizontal son completamente reemplazados usando la técnica de desplazamiento espacial, que además reduce el ruido reflejado ocasionado por el muestreo y mejora la resolución de una señal Y. La técnica de desplazamiento espacial o técnica de 1/2 desplazamiento de imagen se realiza en la dirección horizontal, entre las señales G y R o B. Por tanto el ruido reflejado contenido en las señales G y R o B es de fase inversa. Cuando la señal G se añade a las señales R y B, el ruido reflejado se cancela y cuanto menor es la saturación de la pantalla, mayor es la cancelación del ruido (para los objetos blancos y negros, esta técnica produce alta resolución).



Las bondades del CCD respecto a los tubos se muestra en el cuadro.

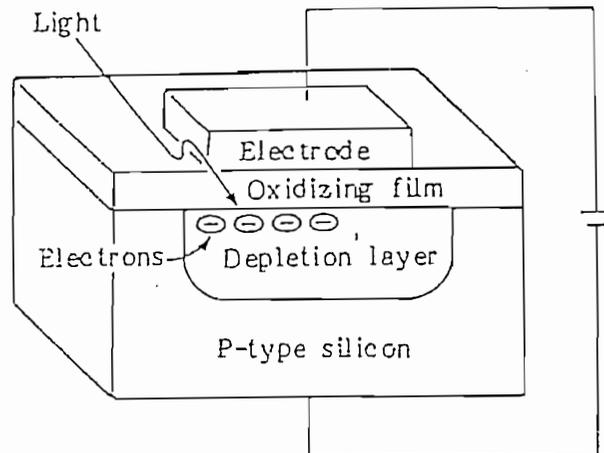
Comparación entre Sensores de imagen de estado sólido y Tubos de imagen.

<u>Función</u>	<u>Sensor de imagen</u>	<u>Tubos usados</u>
Vida y rehabilitación	posibilidad de vida larga	deterioro por calor de radiación beam
Pre-imagen y saturación de imagen (quemar)	no pre-imagen no quema la señal	es inevitable por características fotoelectricas
distorsión geométrica	el arreglo regular de los pixels y la exploración propia da un patron geométrico muy bueno	es difícil la exactitud de haz electrónico por ello existe distorsión geométrico
Resistencia mecánica	es fuerte la estructura del semiconductor	débil por el vidrio filamento tubo

Rápidez de aparición de imagen	es rápida pues no requiere precalentamiento del semiconductor	requiere tiempo de precalentamiento del filamento
Dimensión y peso	Compacto y liviano	distancia para la descarga del beam y espacio para deflección y bobina de foco
Usos de campos eléctricos y magenéticos	no influye	haz electrónico si es influido
Consumo de potencia	bajo consumo de potencia por ser un semiconductor	alto consumo de potencia por el calentamiento de bobinas y el alto voltaje

1.2.2.- Conversión fotoeléctrica (se genera una carga eléctrica usando luz).

Cuando la luz toca la superficie, se genera una carga eléctrica proporcional a su fuerza. En la siguiente figura se muestra la estructura del "condensador" CMOS.



1.2.3.- Almacenamiento de carga.

Cuando se aplica un voltaje positivo a un electrodo de un condensador MOS, un "pozo de potencial", aparece en la superficie del silicón tipo P ubicado bajo el electrodo una carga eléctrica puede almacenarse en ese pozo de potencial.

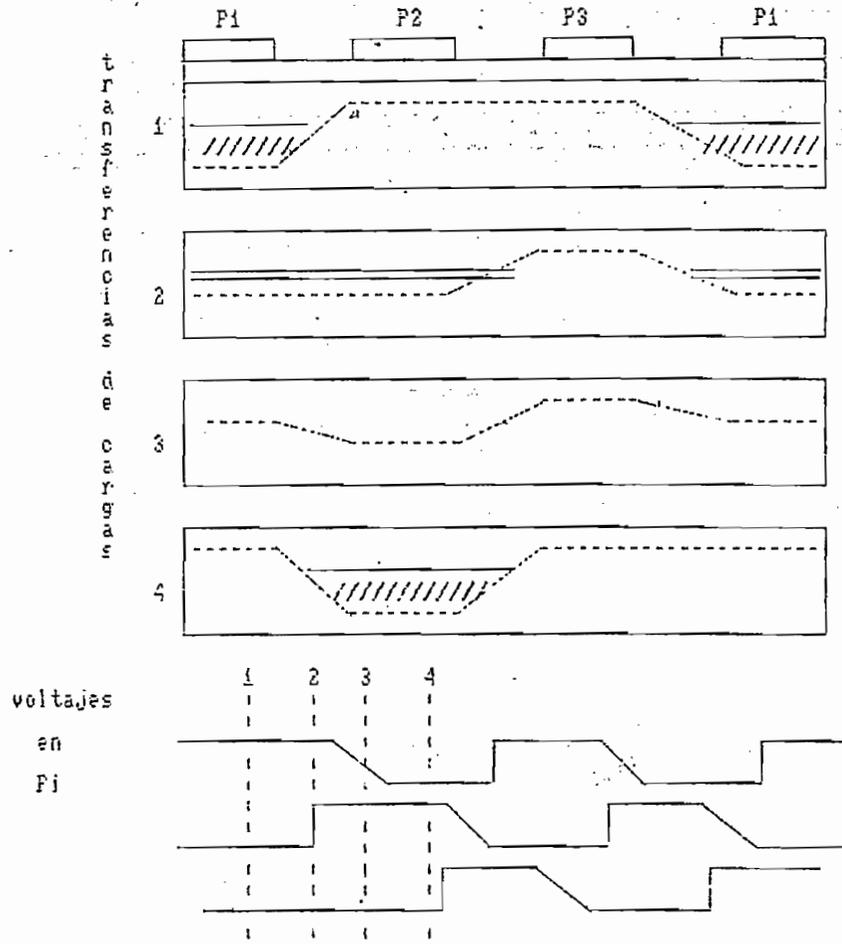
La carga eléctrica que se genera por la conversión fotoeléctrica se almacena en el pozo por un cuadro o por un campo (1/60 seg en NTSC). La carga resultante es entonces leída del dispositivo CCD utilizando una función de transferencia por corrimiento de cargas.

1.2.4.- Operación de transferencia de carga (Se transfiere una carga eléctrica).

La línea de condensadores MOS, es decir, el CCD, tiene una función de transferencia de carga. Cuanto mayor sea el voltaje positivo que se aplique al electrodo del condensador MOS mayor será la profundidad del pozo de potencial.

Cuando diferentes voltajes se aplican a los electrodos de dos condensadores MOS adyacentes, pozos de potencial de diferente profundidad pueden aparecer debajo de los electrodos, y la carga almacenada se mueve al pozo de potencial de mayor profundidad.

El siguiente esquema ilustra las tres fases del CCD, con sus respectivas formas de onda de los voltajes aplicados a los electrodos.



El patrón fotosensible alternante y el registro de transferencia vertical están conectados mediante un registro de transferencia (registro de desplazamiento). Que no es más que compuertas semiconductoras usadas como pozos de potencial para mover o transmitir la carga de los pixels hacia las compuertas de salida al circuito procesador. El esquema de la operación se describe más adelante.

La luz se envía a la porción fotosensible y se genera una carga proporcional a su brillo.

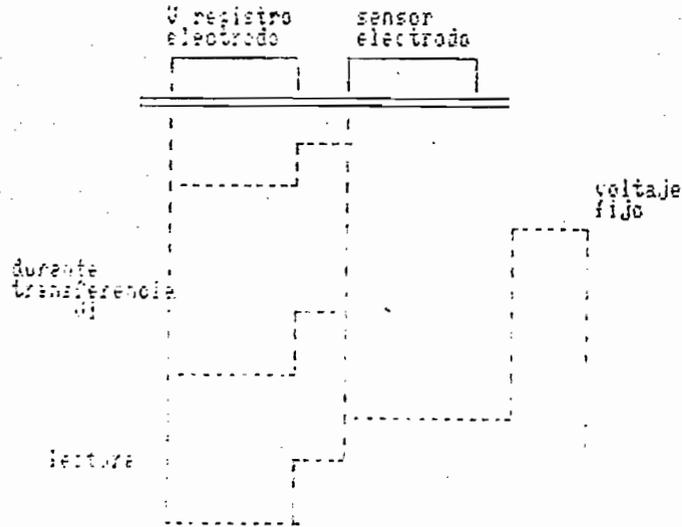
La carga resultante se mueve al registro vertical de transferencia toda a la vez, mezclada con las cargas en dos porciones fotosensibles y secuencialmente movidas al registro de transferencia horizontal. Finalmente, la carga eléctrica se envía a un amplificador de salida y se lee como señal. El fotosensor y el registro de transferencia están separados en el sistema de transferencia interlínea. La carga de señal es transferida secuencialmente al fotosensor, registro vertical y registro horizontal, y luego sale de la sección de salida.

Cada operación se describe posteriormente en forma más detallada las que se considera de mayor importancia para la comprensión funcional del dispositivo en cuestión.

1.2.5.- Operación de lectura del sensor mediante señal impulsora de tres estados.-

Esta unidad usa un sistema de tres estados del registro vertical que elimina la posibilidad de que el sensor miniaturizado envíe parcialmente una señal a otras partes distintas del registro vertical durante la operación como señal de control para el movimiento de las cargas.

En el siguiente gráfico podemos notar claramente los niveles que tendrán tanto los electrodos como los registros de transferencia, para así permitir e impedir que las cargas se muevan, por medio de barreras de potencial que no pueden vencer las cargas por sí solas.



El potencial de el sensor es aumentada por los 11.5 voltios de DC que se aplican al pulso SG constante. El potencial está en el estado de almacenamiento fotosensor al período A1. Al tiempo B1 el voltaje en los electrodos de los registros vertical ha aumentado por lo tanto el pozo de potencial será incrementado.

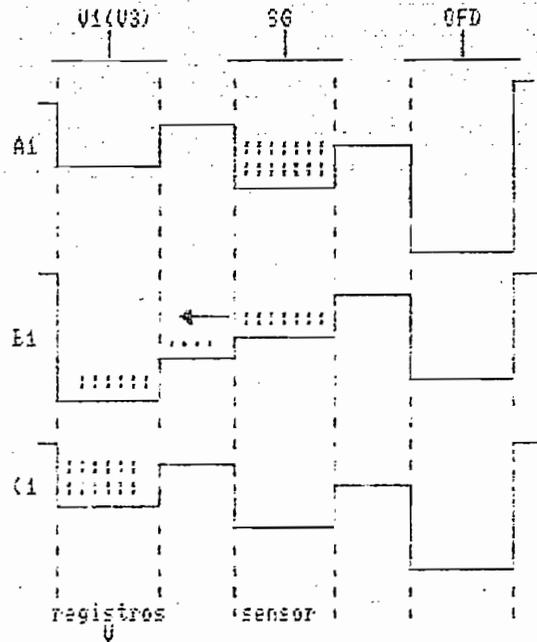
En el instante B1, la carga eléctrica es movida del fotosensor al registro vertical. El voltaje V1 vuelve a disminuir en el periodo C1, por lo que el potencial vuelve al estado de almacenamiento fotosensor.

Las siglas que encontramos aquí representan:

OFD = over flow drenaje (drenaje de sobre-flujo).

SG = sensor gate (compuerta sensora).

V1... (V3) = registros de transferencia vertical.

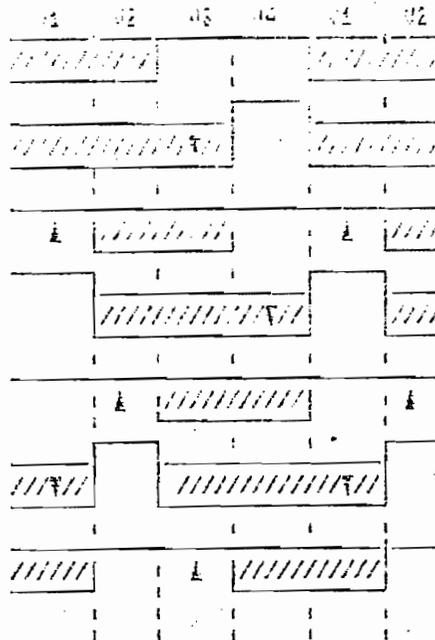


1.2.6.- Impulsor Vertical.-

En este sistema que contiene el gran número de pixels requiere que se reduzca el ancho mecánico del registro vertical. La carga eléctrica máxima se reduce en un sistema de impulsor de dos fases, por lo tanto esta unidad emplea un sistema impulsor de cuatro fases. Entendiéndose por sistema impulsor a la secuencia de voltajes que se aplica a los electrodos de los registros de transferencia para producir el efecto de movimiento controlado de cargas, descrito anteriormente.

Cuando el campo impar o par se lee la carga de dos porciones fotosensibles se mezclan o se transfieren dentro del registro de transferencia vertical y se envían al

registro de transferencia horizontal con una carga eléctrica. Esta lectura, es decir, la transferencia desde el sensor hasta los registros verticales se lo hacen durante cada período de blanking vertical, esto implica que cada $1/30$ de segundo tenemos los pulsos de lectura del sensor.



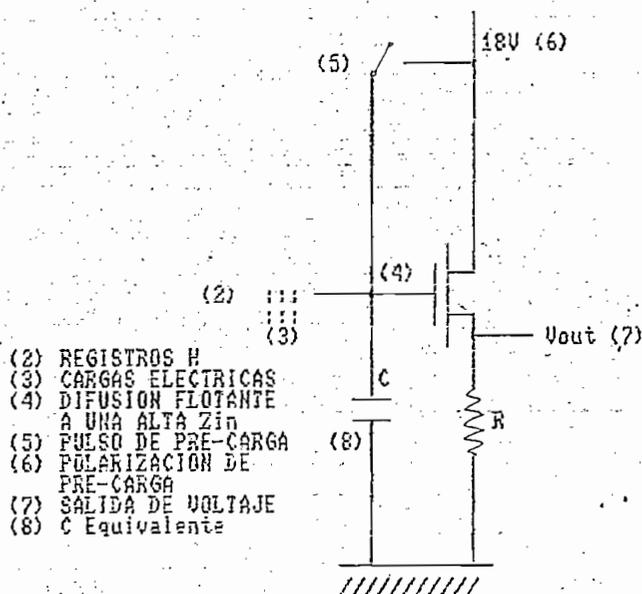
1.2.7.- Transferencia de carga del registro vertical al registro horizontal.-

El potencial H1 del registro horizontal baja (el pozo de potencial se hace más profundo). durante el período de blanqueo horizontal. En este instante, el registro horizontal se alista para recibir una señal del registro vertical. El registro vertical que se detiene en potenciales V1 y V2 bajos durante el período de blanqueo

horizontal, siguiendo la operación de desplazamiento de 4 fases indicada en los pasos, ya mencionada anteriormente, el registro de desplazamiento V4 funciona como la compuerta de salida vertical durante la operación de transferencia y controla la transferencia de carga del registro vertical al registro horizontal (aproximadamente en 11 useg).

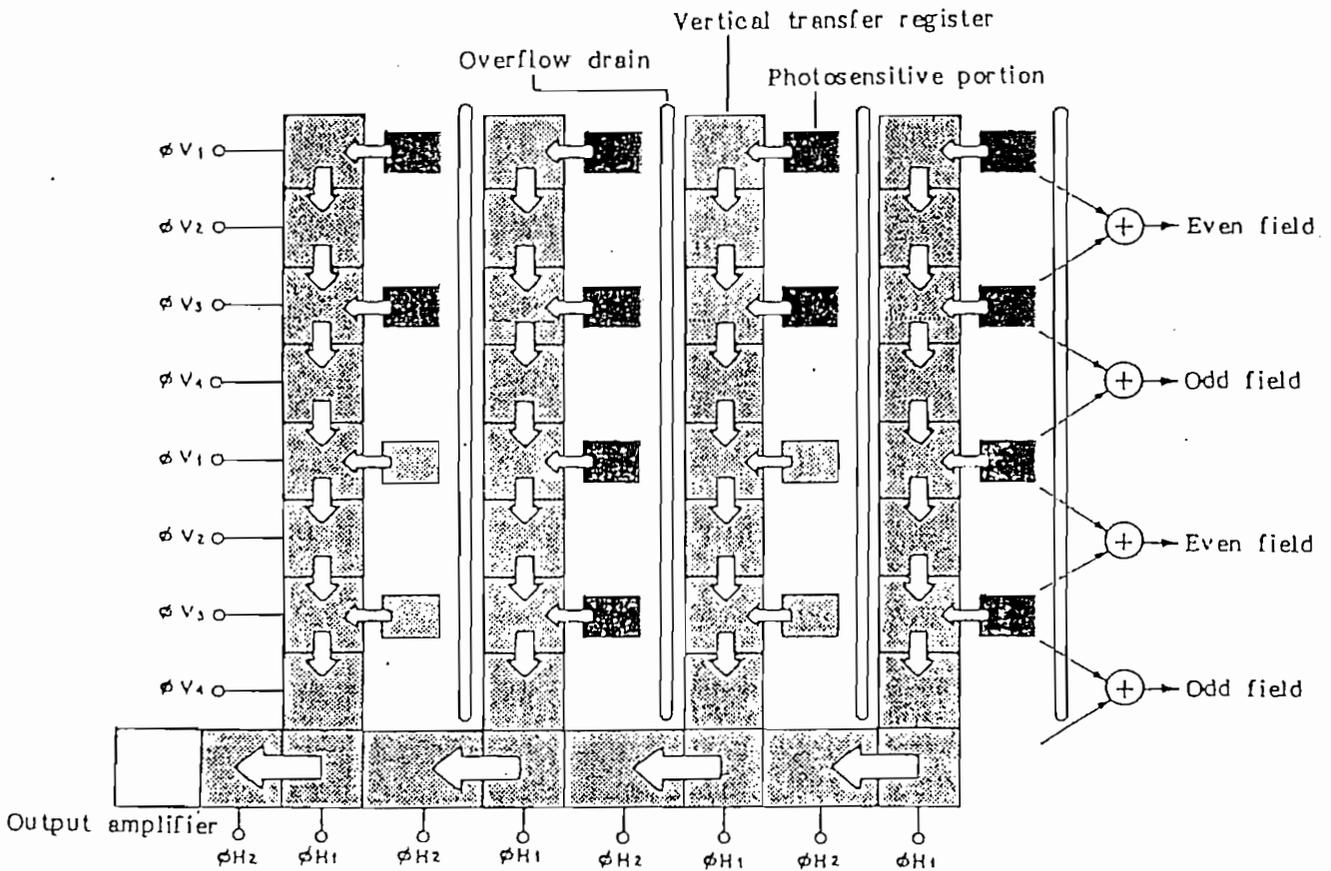
1.2.8.- Mecánismo de transferencia del registro horizontal a la porción de salida.-

La transferencia de carga en estos registros se los hace de la manera como ya se había indicado anteriormente, la innovación en este proceso es que la carga se convierte en voltaje en una relación $V=Q/C$ de acuerdo con la capacidad equivalente C del difusor flotante. Esto se lo hace durante un período efectivo de una línea horizontal, es decir aproximadamente unos 63 useg.

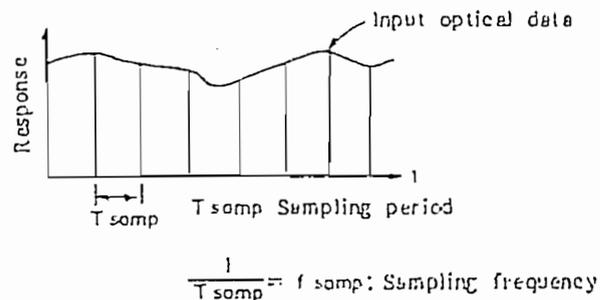


- (2) REGISTROS H
- (3) CARGAS ELECTRICAS
- (4) DIFUSION FLOTANTE
- A UNA ALTA Zin
- (5) PULSO DE PRE-CARGA
- (6) POLARIZACION DE PRE-CARGA
- (7) SALIDA DE VOLTAJE
- (8) C Equivalente

Para poder mostrar ya en una forma esquemática todo el proceso el siguiente gráfico explica: La información captada por los pixels son transferidos a los registros de transferencia vertical en cada blanking vertical, en donde la información será desplazada línea por línea hacia los registros de transferencia horizontal, por donde será desplazada toda una línea; produciéndose la aceptación de línea por línea (semejante a la exploración del haz electrónico de los tubos de vacío)



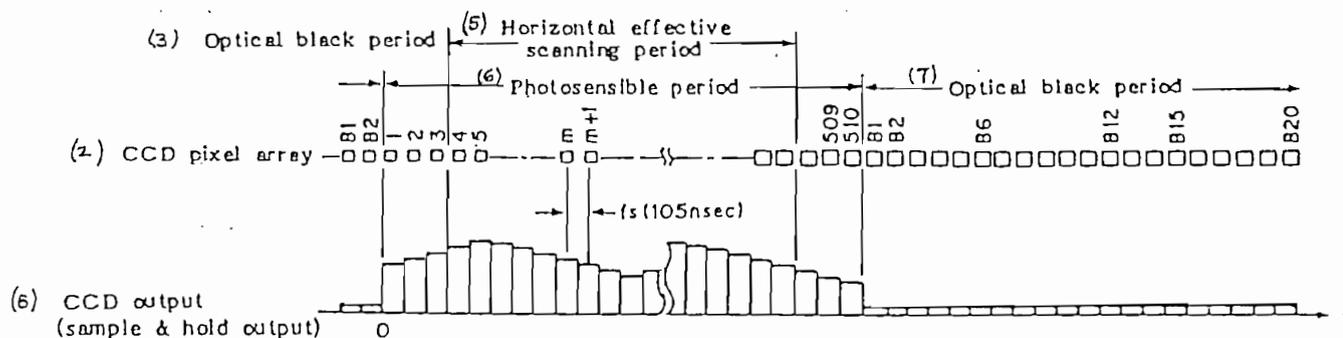
Seguidamente se analiza el tiempo requerido para el muestreo de la señal en este sistema. El sensor de imagen consiste en un arreglo de 250.000 elementos, en los que la información óptica es muestreada espacialmente. Esto se llama un muestreo óptico.



El período de extracción se llama un período de muestra (T_{sample}). La resolución de un sensor de imagen de estado sólido "CCD" está determinado por la frecuencia de muestreo, es decir, de $f_{muestreo} = 1/T_{muestreo}$, y la resolución en el bloque de video de la cámara, está determinado por el número de pixels. La frecuencia requerida se calcula para el circuito integrado ICX018 (NTSC) SONY:

$$f_s = (\text{número efectivo de pixels}) / (\text{tiempo efectivo de barrido})$$

$$f_s = 510 / 55.43 \text{ (useg)} = 9.20079 \text{ MHz}$$



disposición de pixels en una línea de barrido horizontal.

- (1) 510 pixels horizontales.
- (2) 492 pixels verticales.
- (3) Período de negro óptico.
- (4) Arreglo de pixels CCD.
- (5) Período efectivo de barrido horizontal.
- (6) Período fotosensible.
- (7) Período de negro óptico.
- (8) Salida CCD (salida de muestreo).

La información óptica llega a la superficie sensora de imagen en el CCD, es muestreada usando la disposición horizontal y vertical de los pixels. Debido al número de sensores (492x510). se pierde parte de la información, por lo que se requiere más pixels. El CCD lee la información muestreada espacialmente como una función de tiempo, de acuerdo a la operación de auto-barrido.

Este mecanismo de operación se realiza simultáneamente en tres CI-CCD, uno para cada color básicos R-G-B, a los cuales llega la información de su respectivo color por medio de un bloque óptico el cual los difracta separando así la información de color en sus respectivas componentes R-G-B.

1.3.- Criterios para el tratamiento de la señal de luminancia.-

Esta señal es común tanto para receptores en blanco negro como para los de color. Su tratamiento es idéntico en los dos casos.

Luminancia (Y).-

Es la medida de luz de una superficie, sea por generación propia o por reflexión.

Esta definición aplicada a televisión sería el nivel de brillo o luz emitida por la escena, que será captada elemento por elemento (pixel por pixel). Las unidades que se toman es con referencia a la intensidad luminosa, así tenemos:

$$1 \text{ nit} = 1 \text{ candela} / \text{m}^2$$

$$1 \text{ apostilb} = 1 \text{ lumen} / \text{m}^2$$

"Un lumen es la unidad de flujo luminoso, es decir, flujo de luz por segundo que incide sobre un área unidad."

Es por ésta definición que adoptamos las unidades de lúmenes para nombrar a la señal de brillo como luminancia.

Al transmitir esta información desde que se la capta hasta que se reproduce sufre cierta degradación, es por ello que para compensar pérdidas y mejorar la señal en la transmisión, respuestas en frecuencias, alinealidad de los elementos como factores importantes, se hacen varias correcciones en el video que será transmitido.

1.3.1.- Corrección de gamma.-

Un sistema de T.V. debemos considerar que sea lineal, es decir, la luz emitida por la pantalla del Receptor debe ser linealmente proporcional a la luz que incide en la placa que sirve de blanco en la cámara.

Pero la no linealidad de la corriente del haz, con respecto a la característica de la corriente del haz y la luminancia "Y" es proporcional al cuadrado del voltaje en la grilla del Receptor (V_g).

$$Y \text{ proporcional a } (V_g)^2$$

$$Y \approx (V_g)^2$$

en general la relación entre salida de luz y entrada de tensión es :

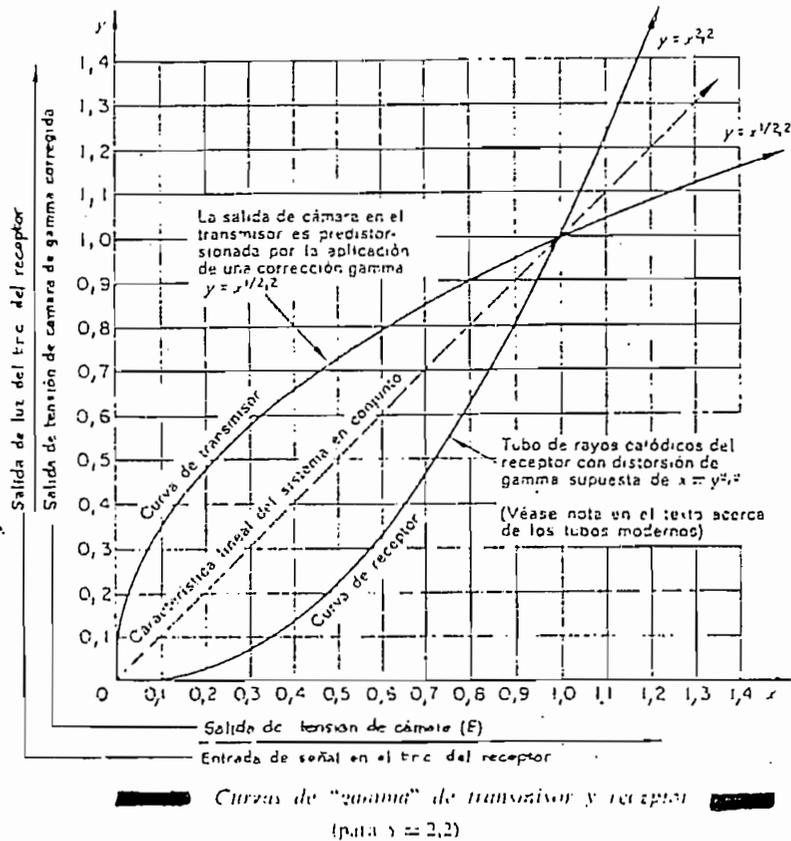
$$Y \approx (V_g)^{\gamma}$$

Compensamos esto en la fuente de imagen con una distorsión opuesta.

$$Y \approx (V_g)^{\gamma} \approx ((E(1/\gamma))^{\gamma}) \approx E$$

donde E es la tensión en la cámara que resulta de una entrada de luz dada.

En el gráfico siguiente podemos observar los efectos que se obtienen, de ahí se deduce lo que debería ser correguido.



Para usos comerciales este valor de gamma se encuentra entre 2.2 y 3.5. El valor aconsejado por la FCC es el gamma y su valor es 1/0.45.

1.3.2.- Fijador de nivel de negro.-

El nivel medio de la señal de T.V. varia según la imagen cambia de brillo, de alto a nulo o blanco a negro. Esta variación de DC es perdida cuando la señal es pasada

por un acoplador RC y amplificada. Para proveer una referencia y reinsertar el nivel de DC, usamos circuitos "clamper" los cuales fijan el nivel de los pulsos de sincronismo a una constante de DC, independiente de las variaciones de nivel del brillo de cada cuadro.

También elimina la interferencia de HUMM que es el ruido causado por interferencias de polarización es decir, la frecuencia de línea (60 Hz) se filtra en los circuitos de procesamiento provocando un "rizado" en la componente DC de polarización. Por el contrario no elimina este humm si el video ha sido tratado con una modulación no lineal.

Las señales que necesitamos generar para producir los colores más comunes que se tienen en una escena son tres, R-G-B (rojo, verde, azul).

Experimentalmente se ha comprobado que:

1-lm blanco = 0.3 lm rojo + 0.59 lm verde + 0.11 lm azul

donde: la abreviación de lumen es lm

Si estos lumenes se los representa como voltajes, llegamos a la ecuación :

$$Y = 0.3 R + 0.59 G + 0.11 B$$

lo cual indica que la luminancia es obtenida también por la adición de los tres colores básicos.

La información de color que se le denomina "Crominancia (Cr)" se sobrepone a la información de brillo (luminancia) para complementar la información de una escena de color, y consta de dos componentes ortogonales (en

cuadratura) denominados I, Q . Por esta razón Cr está determinada por la magnitud y fase, así:

$$Cr = \sqrt{I^2 + Q^2} \quad \theta = \tan^{-1}I/Q$$

Las componentes de crominancia I, Q que están en cuadratura se expresan en función de $R-G-B$:

$$I = 0.60R - 0.28G - 0.32B$$

$$Q = 0.21R - 0.52G + 0.31B$$

Se usan además ejes referenciales $(B-Y)$ y $(R-Y)$ que están a lo largo de los ejes X y Y respectivamente, llamadas también señales diferencia de color, usadas en sistemas de T.V. Europeas. Según estas señales, las componentes I, Q quedan determinadas por las siguientes relaciones:

$$I = 0.74 (R-Y) - 0.27 (B-Y)$$

$$Q = 0.48 (R-Y) + 0.41 (B-Y)$$

Estos ejes I, Q son girados 33° respecto de los ejes x, y (señales $(B-Y)$ y $(R-Y)$ respectivamente); tal que el eje I se encuentre a 123° del eje x , en donde el ojo presenta máxima resolución de color, esto es para matices naranjas y sus colores complementarios azul-verde, denominados también eje naranja-ciano.

Simultáneamente el eje Q estará 33° del eje x , y corresponde a los matices que el ojo es menos capaz de resolver, es decir de mínima agudeza de color, y se lo denomina eje púrpura-amarillo-verde.

Con esta información se consigue tener los colores primarios y complementarios, que se los encuentran en señales patrón, tal como las barras de color cuya presentación es:

blanco/amarillo/ciano/verde/magenta/rojo/azul/negro.

Las caules se generan con combinaciones de los tres colores primarios (R-G-B), con las siguientes proporciones:

	R	G	B
Blanco	1	1	1
Amarillo	1	1	0
Ciano	0	1	1
Verde	0	1	0
Magenta	1	0	1
Rojo	1	0	0
Azul	0	0	1
Negro	0	0	0

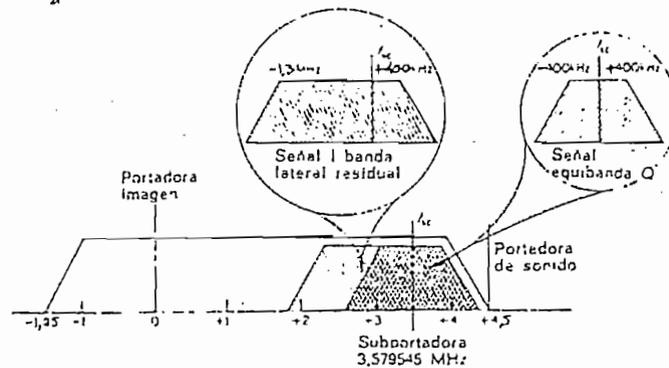
Aplicando las ecuaciones anteriores de Y y C se tiene la distribución estandarizada de barras.

1.4.- Criterios para el tratamiento de Crominancia.-

, La crominancia es la señal que lleva la información del color de cada escena, esta modula a una sub-portadora tal que no haya interferencia con la señal de video generalizada.

1.4.1.- Elección de la frecuencia subportadora de crominancia.-

Es una señal autocontenida y está simplemente añadida a la información normal de imagen, es decir, el detalle de luminancia.



los factores que rigen la elección de la frecuencia sub-portadora de color f_{ec} son:

- a) Para que la interferencia sea mínima f_{ec} debe ser lo más alta posible dentro del ancho de banda del video.
- b) Habiéndose especificado un ancho de banda para cada componente de color que modulan a la sub-portadora, la sub-portadora debe ser lo justamente inferior a 4.0 MHz para que permita la incorporación de las bandas laterales superiores (400 Hz) de las componentes de Crominancia.
- c) La señal de crominancia modula en amplitud.
- d) La diafonía entre la información de crominancia y la de luminancia debe ser mínima.
- e) Puesto que la subportadora cae en el espectro normal de videofrecuencia, podrá pasar por los amplificadores de video en los receptores B/W. Como tiene una alta frecuencia producirá una figura de puntos a lo largo

de cada línea de la trama o cuadro. La relación numérica entre f_{ac} y la frecuencia de línea f_n se debe elegir de modo que la figura de puntos se desplace horizontalmente a en las sucesivas líneas del cuadro de manera que la visibilidad de esta interferencia sea mínima y tal que cumpla con las condiciones antes mencionadas.

En el desarrollo de Fourier se puede notar que existen armónicos de la frecuencia de exploración horizontal f_n (15750 Hz), y es en donde la señal se encuentra concentrada; estas son múltiplos de la f_n es decir, f_n , $2f_n$, $3f_n$...etc. El armónico más alto que alcanza los 4.0 MHz (ancho de banda para una señal de T.V.) será evidentemente:

$$\text{ancho banda}/f_n = 4.0 \text{ MHz}/15750 = 253.96$$

Si a esto se le considera la propia forma de presentación del video es decir, las componentes de 30Hz debidas a las señales de sincronismo de campo, cada armónico de línea horizontal tiene grupos de frecuencias laterales, alrededor de ella a intervalos de 30 Hz.

Dentro de estos espacios (armónicos de f_n y f_{campo}) está acomodada la señal de crominancia.

En el sistema NTSC la subportadora de crominancia, está situada exactamente entre los armónicos de f_n , es decir, su frecuencia estaba desplazada la mitad de la frecuencia de línea con respecto a un armónico de f_n .

Esto es lo que se conoce por desviación de media línea, e implica la elección de una subportadora de frecuencia igual a un múltiplo impar de la mitad de la frecuencia de horizontal.

El número de ciclos de subportadora por línea es igual a f_{ec}/f_h . Pero en el sistema NTSC $f_{ec}=n*f_h/2$ donde n es un número impar. Por tanto

$$f_{ec}/f_h = (n*f_h)/(f_h*2) = n/2$$

Así tenemos $f_{ec}=455*f_h/2$ para el sistema NTSC. Como la señal de crominancia es en si misma una portadora de r.f. que está interrumpida al ritmo de la frecuencia horizontal, tiene un espectro constituido por aglomeraciones de frecuencia horizontal espaciadas a intervalos de f_h . Estos grupos o aglomeraciones estan situadas a ambos lados de f_{ec} .

La subportadora hace que el brillo de una línea de cuadro aumente y disminuya senoidalmente desde un extremo al otro; si se consideran a los picos positivos como puntos blancos y a los picos negativos como puntos negros la impresión de la imagen total de puntos (imagen graneada o porosa) que es muy molesta para el observador, dando no una proyección de una imagen continua, como psiquicamente se persive a través del ojo humano.

Puesto que n es un número impar, hay un número de semiciclos en cada línea. En dos líneas cualesquiera sucesivamente exploradas de un campo, los puntos blancos se

mueven a lo largo de la línea durante un semiciclo. Verticalmente esto tiende a producir la cancelación, ya que los puntos blancos y negros caen uno debajo de otro.

Como se debe preservar la información de fase de la subportadora para la demodulación del sincronismo, unos pocos ciclos de la frecuencia de subportadora es adherida inmediatamente después del pulso de sincronismo horizontal. En el receptor, el burst es pasado a través de un circuito de disparo (gating) que sincroniza periódicamente a un oscilador a cristal al final de la exploración de cada línea. Este oscilador se utiliza para sincronizar al demodulador y así decodificar la información de color.

La frecuencia de color en el sistema NTSC es de 3.579545 MHz +/- 10Hz. Esta frecuencia tiene relación con dos frecuencias requeridas. Cuando multiplicamos por $2/455$ se obtiene la frecuencia de exploración horizontal de 15734.3 Hz y la frecuencia vertical multiplicando por $4/(455 \times 525)$ dando los 59.94 Hz. Valores cercanos a los teóricos, tal que los receptores de B/W no tengan problemas para su sincronización, pues los circuitos aceptan en un rango de tolerancia del 1%. En adición al batimiento de frecuencia entre 3.579545 MHz y la portadora de audio 4.5 MHz (4.25 MHz el límite inferior del ancho de banda de audio) es también intercalada para que el batimiento y la interferencia sea mínima.

Determinada la sub-portadora de color (subcarrier); se analiza el comportamiento general de una modulación A.M. y especialmente en cuadratura, Q.A.M. para así determinar las características de esta señal de crominancia.

En previsión de un concatenamiento del texto hay que destacar los siguientes puntos:

- a) Las componentes de color I, Q son simplemente tensiones que pueden ser positivas, negativas o cero, en un instante durante una línea de exploración.
- b) En la señal de imagen transmitida, las dos señales I, Q de color tienen que modular a dos señales de radio frecuencia que serán detectadas en el receptor.
- c) Los diagramas vectoriales de señal de crominancia muestran las componentes de modulación de r.f. correspondientes a las componentes I, Q del color original que dan lugar a la crominancia.

Así por ejemplo si tenemos $(R' - Y') = 0.4$ y $(B' - Y') = 0.2$; formarán las señales $I=0.242$ y $Q=0.274$. Estas son tensiones y por lo tanto no se podría dibujar un diagrama vectorial para sumarlas y construir la resultante. Ambas son solamente tensiones constantes, sin embargo sí se realiza un proceso de modulación con subportadoras que se puedan diferenciar, y si estas sub-portadoras están desfasadas 90° entre sí (cuadratura), entonces las dos señales r.f. de amplitud modulada y portadora suprimida pueden ser representadas por dos vectores que formen un ángulo de 90°

entre sí, y si luego se suman estas dos señales moduladas los dos vectores se pueden combinar y formar un vector resultante cuyo ángulo será mayor a 0° y menor a 90° con respecto a ellas.

d) Los diagramas vectoriales de televisión en color son diagramas de r.f. Se refieren a la señal modulada de crominancia y a sus componentes.

Propiedades más importantes de las señales de crominancia

QAM.-

- a) La señal de crominancia es la combinación de dos señales moduladas separadas.
- b) Cada una de ellas es señal de r.f. de amplitud modulada con doble banda lateral y portadora suprimida.
- c) Las portadoras son de 3.579545 MHz desfasadas 90° .
- d) Cuando dos señales separadas con modulación a.m. se suman para formar una señal de crominancia, las características individuales no se pierden ya que en cualquier instante la amplitud y la fase de la señal única de crominancia que forman, está totalmente determinada por las amplitudes y las fases de las dos componentes.
- e) En información B/W las señales diferencia de color y las componentes I, Q son cero desapareciendo la señal de crominancia.

Señales de a.m. con portadora suprimida

Cuando una onda es modulada en amplitud por una sola onda senoidal de frecuencia f_m Hz la señal resultante tiene tres componentes de frecuencia: f_c , $(f_c + f_m)$, $(f_c - f_m)$ Hz respectivamente llamadas frecuencias laterales.

Como cada componente de frecuencia moduladora tiene sus correspondientes frecuencias en ambas bandas laterales, se deduce que la información total sometida a proceso en un sistema a.m. puede ser enviada transmitiendo ya sea una o ambas bandas laterales, sin la portadora. No obstante, para recuperar la información en el receptor es necesario tener una muestra de esta portadora, es por ello que se inserta el burst.

$$(f_c + f_m) - f_c = f_m$$

$$f_c - (f_c - f_m) = f_m$$

Las señales de imagen de televisión utilizan la banda lateral, es decir tienen una banda lateral completa y una banda lateral suprimida parcialmente. El sistema de doble banda lateral con portadora suprimida ha encontrado su aplicación más útil en los sistemas de televisión a color NTSC para la señal de crominancia. Esto por las siguientes razones:

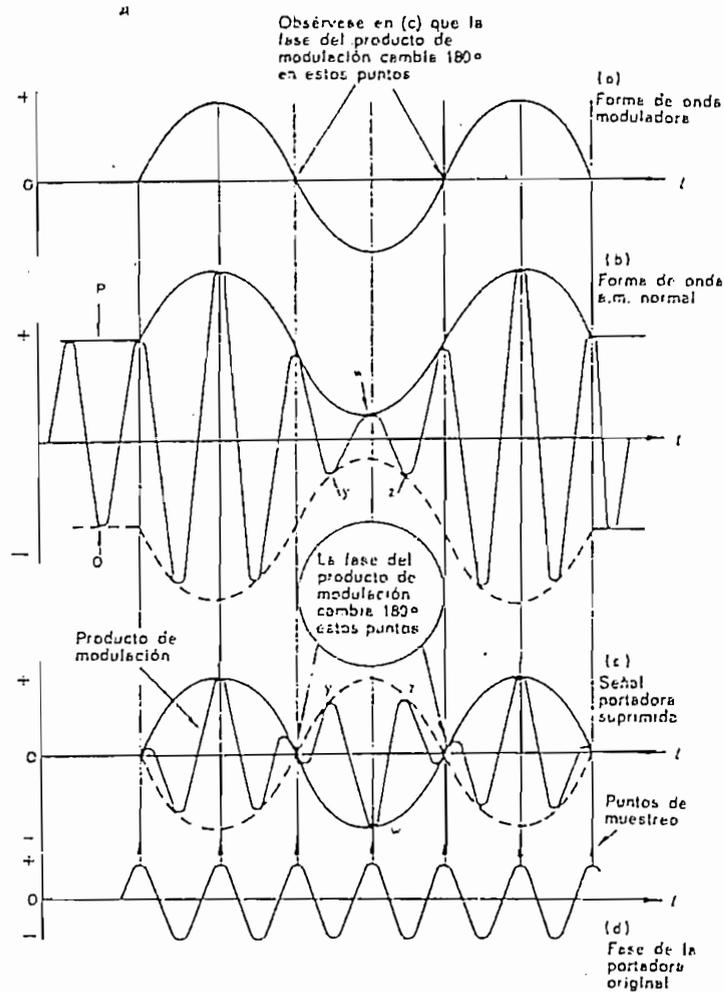
- a) Las señales de doble banda lateral con portadora suprimida permite identificar fácilmente las señales I, Q de color con valores positivos y negativos.

- b) Minimiza la interferencia producida por la señal de crominancia en los receptores B/W cuando reciben programas de transmisión en color.

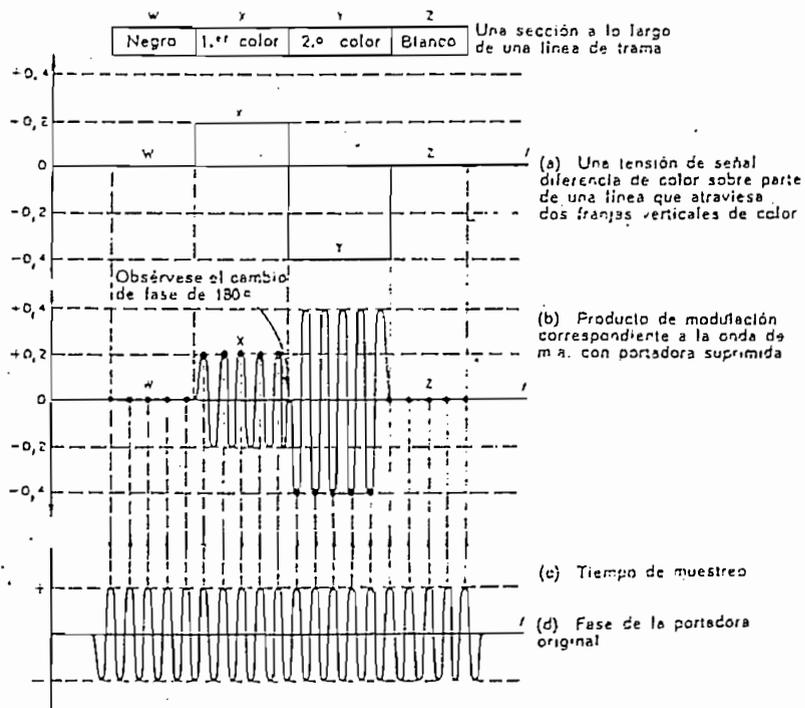
Pues la mayor parte de la potencia de una señal con modulación a.m. normal está en la componente de portadora, y para profundidades normales de modulación solo queda una energía relativamente pequeña donde realmente se necesita, es decir, en las bandas laterales que contienen la información. La relación de potencia de banda lateral a la potencia de portadora aumenta con la profundidad de modulación, e incluso con la máxima profundidad (es decir modulación del 100%); dos tercios de la potencia total están en la portadora y la potencia útil de banda lateral es solamente un tercio. Como hay tanta energía en la componente de portadora, cuando se la suprime, se elimina mucha potencia de interferencia. Además de esto, las componentes I, Q de color que constituyen la información de modulación son cero cuando el detalle de la imagen no es de color, por lo que en este caso las bandas laterales también desaparecen, no quedando componentes de la señal de crominancia en toda la señal de video.

El siguiente gráfico nos indica un bosquejo de una señal con modulación a.m. y portadora suprimida.

.../...



Así por ejemplo; en una señal al azar que tiene tensiones o información de algunas barras de color y su cambio de positivo a negativo nos indicará que no se pierde información de este hecho aún cuando se transmite con portadora suprimida, de esta manera tenemos la optimización de la transmitir la señal inteligible y sin desperdicio de potencia.



En X el producto de modulación tiene una amplitud que es directamente proporcional a una de las componentes de color. Si comparamos con la fase de la portadora veremos que se encuentra en fase con esta señal, ya que la componente de color es positiva. Recíprocamente en Y cuando la polaridad de la componente de color es negativa en lugar de ser positiva, el producto de la modulación está en oposición de fase con la portadora suprimida original. El cambio de fase de 180° tiene lugar en la transición de X a Y.

El muestrear de esta forma permite leer una sucesión de tensiones iguales en toda la anchura de un intervalo

uniforme de la información de color, por ello el circuito de salida del demodulador incluye un filtro pasa bajo que alisa las tensiones sucesivas de salida (que son impulsos a la frecuencia de subportadora) para que produzca una tensión constante que representa la tensión original de una componente de color.

A fin de transmitir tanta información de color como pueda percibir el ojo, pero no más, a la señal I se le ha asignado una banda más ancha que a la señal Q.

En el sistema de 525 líneas de U.S.A. el ancho de banda lineal de la señal de video es 4 MHz o 4.2 MHz medido hasta el punto en que la respuesta desciende 2 db.

- a) Componente Q de la señal de crominancia: ancho de cada banda lateral es de 400 kHz.
- b) Componente I de la señal de crominancia: ancho de la banda lateral inferior es de 1.3 MHz, y el ancho de banda lateral superior es de 400 KHz.
- c) Información de luminancia con ancho de banda de 4.0 MHz.

Según esta clasificación las áreas grandes y rasgos prominentes de la imagen son resueltos plenamente en sus verdaderos colores. Los detalles más finos de color resueltos plenamente corresponden a frecuencias de video hasta el orden de 400 kHz y los detalles finos del color azul-verde es resuelto a un límite de 1.3 MHz, los

detalles más finos son solo resueltos en B/W.

La frecuencia de subportadora de color está situada 400 kHz por debajo de 4.0 MHz aproximadamente. La componente Q es radiada como señal de doble banda lateral equibanda, mientras que la señal I es de la forma de banda lateral residual con una banda lateral superior de 400 kHz y una inferior de 1.3 MHz.

1.5.- Tratamiento de Sincronismos.-

En todo fenómeno físico parte de su comportamiento es el sincronismo con todo el sistema; o secuencias que son relacionadas una a otra a través de estos.

Para obtener esta armonía requerimos de varios pulsos en diferentes ubicación de la secuencia.

1.5.1.- Intervalo Horizontal.-

La porción horizontal de la forma del video compuesto consiste de líneas horizontales, cada una contiene la información de video y pulsos de sincronismos, los cuales provee del blanking, tiempo de retorno y otras informaciones. La descripción de los eventos internos en una línea en el blanking horizontal nos da un conocimiento del propósito de esta forma de onda.

El front porch previene de informaciones erróneas (voltajes) de video por switcheos prematuros de los

circuitos de sincronismo. El oscilador de barrido horizontal es refijado por el flanco anterior del pulso de sincronismo, mientras que el frente del pulso de sincronismo es una referencia para retornar el nivel de DC en los circuitos del receptor.

El back porch provee el tiempo para el blanqueo del haz electrónico de la pantalla. Durante la transmisión de color, se adhiere un 3.58 MHz sinusoidal (unidad de burst empacutado) en el back porch para dar la información de la fase del color, del cuadro que se esta transmitiendo. La información del video está contenido entre el fin del back porch y al comienzo del front porch.

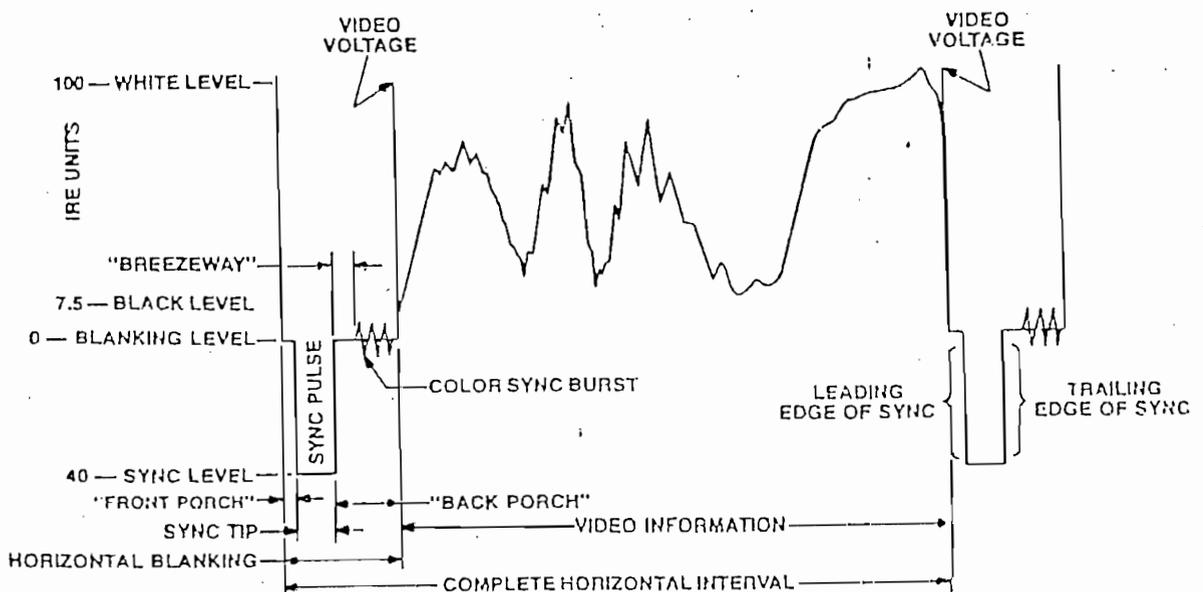
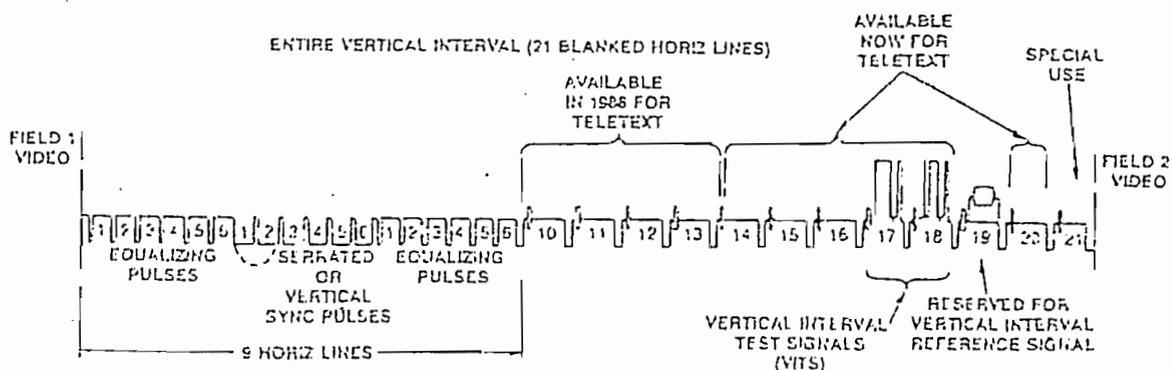


Fig. 4. Horizontal line.

1.5.2.- Intervalo vertical.-

En este instante, los pulsos existentes son muy importantes; está compuesta por algunas líneas horizontales que no son muestreadas en la pantalla y en ellas se mandan pulsos, señales de prueba, fuentes de códigos de identificación, señal de referencia, encabezamiento y teletex.



Los primeros seis pulsos, llamados de pre-equalización sincronizan para que no haya saltos en la información de video en campo 1 y campo 2. Estos pulsos ocurrendoblemente en el tiempo de un sincronismo horizontal (periodo = 63.5 us, $f = 15750$ kHz) y asegura que el disparo vertical ocurra en el tiempo preciso para campo par e impar.

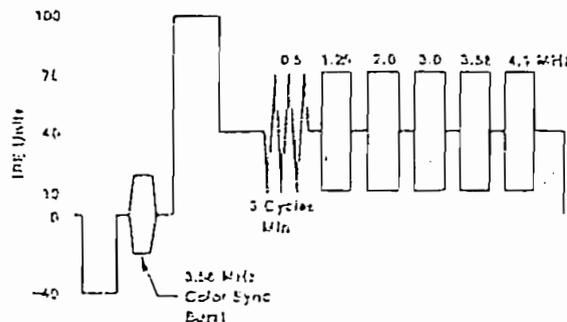
Los seis pulsos dentados verticales que retorna al haz explorador a la parte superior de la pantalla, seguidos de seis pulsos de pos-equalización, los cuales aseguran la regularidad e identificación de los diferentes campos.

Recientemente han sido ampliados los usos de las señales de prueba, como veremos una de estas aplicaciones.

1.5.2.1.- Señal de prueba en intervalo vertical (VITS).-

Esta señal de VITS es usada para evaluar varios parámetros del sistema de radiación de la señal su mejoramiento y el monitoreo diario. Los VITS son transmitidos durante la operación activa para asegurar la calidad de continuidad y precisar en términos de color y distorsión. Considerando el mejoramiento para la transmisión, la calidad de los VITS frecuentemente determina si un problema es inminente que suceda en el sitio del transmisor.

La señal de VITS, Multiburst, cuyo objetivo es dirigido específicamente a ganancias de frecuencias, además fija bien al monitor y selecciona compresión de frecuencias. Este patrón de prueba usualmente consiste de una barra blanca (100 IRR) seguido por 6 rafagas de ciclos senoidales de frecuencias 0.5, 1.25, 2, 3, 3.58, 4.1 MHz.



1.5.3.- Burst.-

Para empezar a precisar sobre la señal de sincronismo de color, indicaremos parte de la utilidad de insertar el burst en todo el sistema de televisión para transmisión.

1.5.3.1.- Señal de Sincronismos en NTSC.-

El grupo-empaquetado " burst " retarda las características especificadas para el transmisor introduciendo un retardo extra de 170 ns para la luminancia y señal de sincronismo, relativo para la portadora de color burst. El breezeway es especificado en 379 ns mínimo, esto en la salida del transmisor, pero deberá ser más grande por lo menos en 170 ns que el de la salida de un SPG. Para un buen diseño de ingeniería requiere que el tiempo de subida de los pulsos de SPG sean algo más rápidos que los especificados en las reglas del FCC.

Esto es para controlar la forma de la transición del pulso de sincronismo producido por SPG usando filtros con función de transferencia de senos cuadrados. Estos son fases equalizadas, filtros pasa bajos que limitan el espectro de frecuencias de estos pulsos. Una ventaja particular en un limitador de banda de la salida de un SPG es que reduce el ruido metálico y la diafonía.

Esto es especialmente importante en la unidad SPG, en la cual el tiempo de subida debe ser tan corto como 10 ns. Los filtros de seno-cuadrado tiene una frecuencia de corte

de aproximadamente al inverso del tiempo que le tome llegar a la mitad de la amplitud del pulso. El tiempo de subida es siempre medido desde 10% hasta 90%.

El burst no es transmitido durante las primeras 9 líneas en cada blanking vertical. Las nueve líneas no transmitidas con burst es producido en el SPG por la inhabilitación de generar los pulsos de burst-flag, por ello podemos decir que la ausencia de los 9 paquetes de burst es aparente.

La bandera (burst-flag) precede al burst, para acomodar el retardo en el filtro pasabanda a través del cual el burst flag y burst debe pasar en el encoder.

1.5.3.2.- Técnicas de black burst:-

El costo para distribuir las salidas individuales de un SPG en toda la instalación de T.V. por múltiples cables coaxiales han sido determinantes para el desarrollo de la señal de black-burst.

Esta señal lleva toda la regulación requerida de la información de un SPG hacia otros SPG pudiendose instalar sistemas maestro-esclavo. El esclavo SPG puede así tomar el tiempo relativo para un punto común con la planta, tal que encodifique la señal de video de una fuente y esté en preciso sincronismo con respecto a otro.

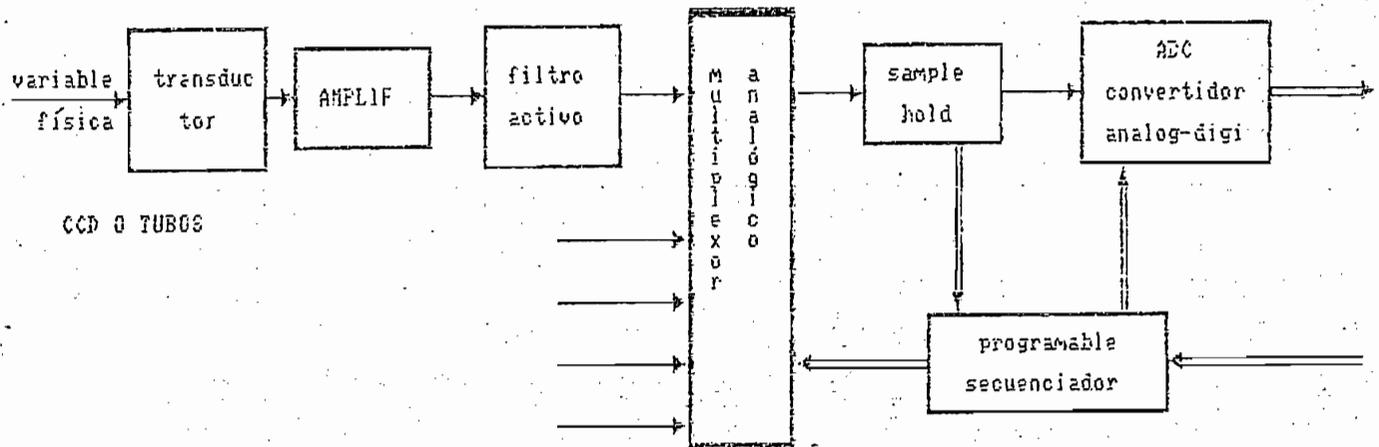
Esta condición asegura fases iguales, que es esencial durante el uso del switcher en fade, wipe y disoluciones

CAPITULO II

2.- Fundamentos teóricos para la digitalización de una señal de video.-

2.1.- Conversión Análogo-Digital.-

El diagrama de bloques general se puede decir que es el siguiente:



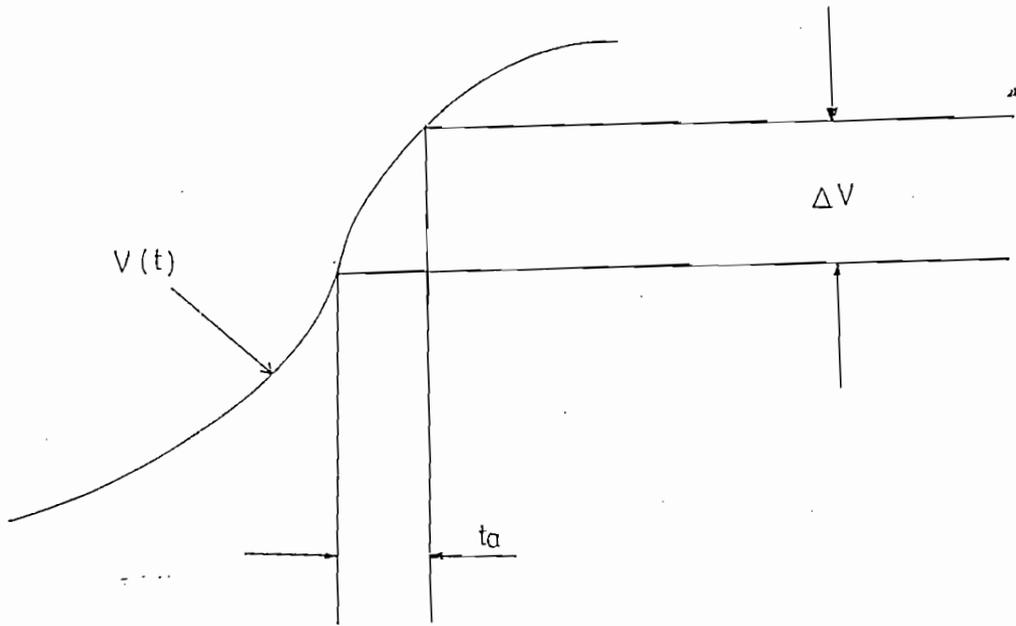
La captación de la variable física (imagen), su amplificación y procesamiento analógico no es parte de este tema (se lo obtiene ya a la salida de una cámara o de un transmisor analógico cualquiera). Los filtros Activos pasabajos, se utilizan para eliminar las componentes de alta frecuencia o ruido de la señal; puede ser necesario también realizar algunas operaciones no lineales sobre la señal tal como multiplicación por otra función para deslinealizar, etc.

El convertidor A/D realiza las operaciones de cuantización y codificación de una señal en una cantidad finita de tiempo; este depende tanto de la relación del convertidor como del método de conversión particularmente usado. La velocidad de conversión necesaria en una situación particular depende del tiempo de variación de la señal a ser convertida y de la cantidad de resolución deseada. Al tiempo empleado para hacer una medición o conversión, se lo conoce como "tiempo de apertura".

Al tiempo de apertura se lo considera como incertidumbre de tiempo e influye en la incertidumbre de amplitud. Ambas están relacionadas por la relación de tiempo al cambio de señal.

{Gráficamente podemos indicar lo siguiente:

.../...



Para el caso particular de la conversión de una señal sinusoidal, la relación máxima de cambio ocurre en el cruce por cero de la onda y el cambio de amplitud es:

$$\Delta V = [d(V \text{ Sen } \omega t)/dt]_{t=0} + t_a = V \cdot \omega \cdot t_a$$

$$\Delta V/V = 2\pi f \cdot t_a$$

Del resultado podemos determinar, por ejemplo, el tiempo de apertura (o tiempo de conversión) requerido para digitalizar una señal de 1 KHz con 10 bits de resolución, esto es una resolución de una parte en 2 a la 10 o 0.1%.

Utilizando la ecuación anterior:

$$t_a = (\Delta v/v) \cdot (1/2\pi f) = 0.001/(6.28 \cdot 1000) = 160 \times 10^{-9} \text{ (seg)}$$

Esto implica que se requiere un tiempo de apertura de solo 160 nanosegundos para mantener en 1 bit (0.1%) de resolución debido al cambio de la señal entrante. De este modo puede verse que para convertir aún señales de

variación muy lenta con niveles de resolución moderado se requiere un convertidor A/D extremadamente rápido y por lo tanto caro. Para resolver este problema se utiliza un circuito de "muestreo y retención", que reduce considerablemente el tiempo de apertura, al tomar una muestra de la señal en un tiempo muy corto y manteniendo este valor durante el tiempo requerido para hacer la conversión. Actualmente los conversores A/D vienen ya con esta innovación del Sample-Hold; resolviendo así unos problemas y otros tan solo disminuyendolos.

Los sistemas que implican la transmisión de señales digitalizadas y codificadas se denominan sistemas de modulación por codificación de pulsos (PCM).

Hay muchas ventajas en la utilización de los sistemas de este tipo:

- 1- Las señales pueden regenerarse periódicamente durante la transmisión; puesto que la información ya no se encuentra contenida en la amplitud continua variable de los pulsos, sino que consiste en símbolos discretos.
- 2- Toda clase de circuitos digitales puede emplearse durante la totalidad del procesamiento.
- 3- Las señales pueden ser procesadas digitalmente según convenga.
- 4- El ruido y la interferencia pueden ser apropiadamente minimizados mediante códigos, etc.

En el proceso de la digitalización de una señal analógica se debe cuantizar la señal y es la subdivisión de las amplitudes de una señal en niveles predeterminados. Al usar este procedimiento no podemos reconstruir la señal analógica original tan fiel; pero no es un problema tan grave, pues el ruido introducido tanto en el circuito procesador y transmisor como el ruido en el canal de transmisión; hace que en el receptor, el detector no sea capaz de distinguir entre variaciones muy pequeñas y finas de la señal.

Aunque la separación entre niveles que se muestran aquí son uniformes, con frecuencia en la práctica la separación de niveles para cuantizar no son uniformes; se los hace intencionalmente no lineales; disminuyendo el espaciamiento en valores bajos de amplitud para mejorar así el comportamiento del sistema al ruido. A esto se lo llama compresión; y según esta cuantificación serán codificadas.

Para el proceso de codificación, una de las dificultades que se presentan con la conversión normal de decimal a binario es que, cuando cambia un dígito decimal a otro adyacente, en el código binario cambia más de un dígito binario, acrecentando así la posibilidad de error en una transmisión, por ejemplo, en el cambio decimal de 3 a 4 y 7 a 8. Lo mejor sería tener un código que tenga un cambio a la vez cuando varía de un nivel a otro, es decir, de un

dígito decimal a otro adyacente. Este código podría ser el código Gray, que a su vez no deja de ser PCM.

En la transmisión y procesamiento de la señal de video digitalizada se tiene que analizar exhaustivamente el ruido y su relación con el circuito, es decir, la relación señal-a-ruido (S/N o SNR), además de la compresión en PCM.

El ruido de cuantización puede reducirse hasta cualquier grado que se desee escogiendo los pasos de cuantización, lo más próximos posibles; pero hay aquí un problema de compromiso, cuanto mayor es el número de pasos cuanticos empleados tanto más grande es el número de dígitos binarios o bits que se necesitan para representar la señal y, por lo tanto, el ancho de banda necesario para la transmisión será también mayor. Se escogen entonces normalmente tan pocos niveles de cuantización como sea posible para mantener la consistencia con los objetivos de la transmisión. Ya se ha mencionado que para la transmisión de voz por medio de PCM algunas pruebas subjetivas indican que 128 niveles, o sea PCM de 7 bits son suficientes para asegurar una alta calidad en la transmisión.

El ruido de cuantización, se calculará, por el promedio cuadrático o su raíz cuadrada del nivel del ruido rms, pero se le relacionará con alguna medida del nivel de la señal; su valor cuadrático medio o a su valor pico cuadrático, para determinar el rendimiento respecto al ruido de cuantización de un sistema PCM.

El siguiente método tiende a disminuir este error de cuantización, en donde los niveles de cuantización no son uniformes, para adaptar las características de la señal y son comprimidas en forma logarítmica, y en el receptor la señal se expande a su amplitud original por medio de una forma antilogarítmica. La combinación de los procesos de compresión y expansión se conoce con el nombre de compansión para abreviar.

Para comparar estos dos métodos y sus rendimientos notemos:

El ruido de cuantización con espaciamiento del mismo nivel se tiene que es:

$$S_o/N_o = 3M^2 \quad M: \text{número de niveles empleado}$$

$$M^2 = n^{2m} \quad m: \text{número de pulsos del grupo de codificación (bits)}$$

$$S_o/N_o = 3n^{2m} \quad n: \text{número de niveles de codificación (n=2 --> bits)}$$

si $n=2$ código binario

$$(S_o/N_o)_{dB} = 4.8 + 6m$$

para un sistema de 128 niveles, es decir 7 bits

$$S_o/N_o = 47 \text{ dB}$$

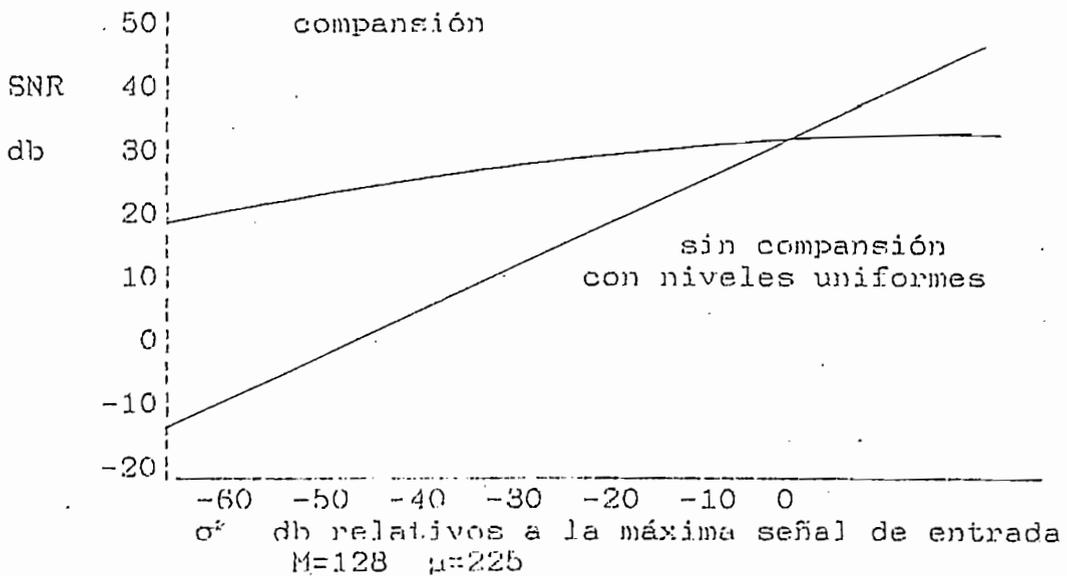
Relación señal a ruido con compansión

Al analizar este tipo de SNR se toma como ejemplo más representativo a un sistema usado en Europa y America. Una de estas es la denominada ley μ cuya expresión SNR relaciona a la potencia cuadrática media de la señal, dividida para el

error cuadrático medio $E(\epsilon^2)$.

$$\text{SNR} = \sigma^2 / e(\epsilon^2) = 3M^2 / [\ln(1+n)]^2 [1 + 2E(|x|) / u\sigma^2 + 1/u^2 \sigma^2]$$

Cuando $u \rightarrow 0$ entonces $\text{SNR} = 3M^2 \sigma^2$ que es la expresión encontrada para SNR con espaciamiento uniforme, cuyas demostraciones son largas y no propias del tema. El siguiente gráfico nos indica en forma clara las ventajas de esta técnica.



Esto indica que para cada ley de compresión logarítmica se obtendrá gráficos similares, como por ejemplo tenemos 2 leyes más usadas:

ley u : usada en Norteamérica (EE.UU.) y Japón

$$Y = \ln(1+ux) / \ln(1+u) \quad 0 \leq x \leq 1$$

$u=255$ 15 segmentos (audio)

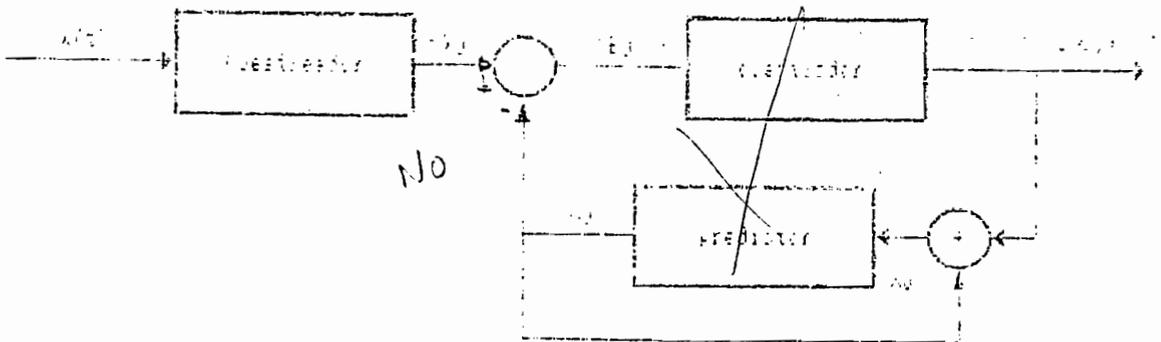
ley A: usada en Europa

$$Y = Ax/(1+\ln A) \quad 0 \leq x < 1/A$$

A=87.6 13 segmentos (audio)

El análisis realizado hasta aquí para audio nos servirá para poder entender las características en la modulación DPCM (modulación por codificación de pulsos diferencial) que es la más usada en el tratamiento que se le dará a una señal de video, en tal aplicación existe un alto grado de redundancia en la información que se va a transmitir. Así la información pasada contiene también parte de la información actual, como para poder predecirla aceptablemente bien, de manera que en las señales nuevas solo necesitan ser transmitidas las variaciones, esto sí se producen cambios significativos en ellas.

Así tenemos:



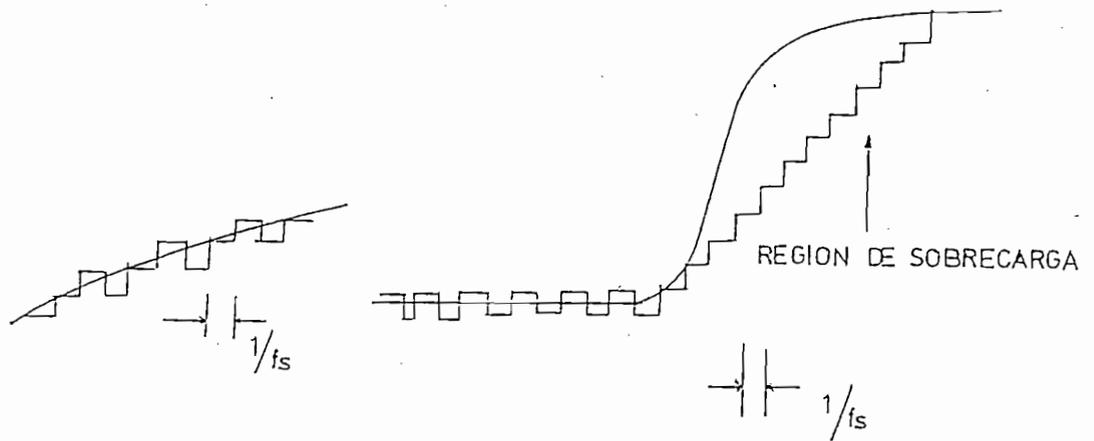
El predictor empleado es generalmente una suma "ponderada de un cierto número de muestras pasadas y estimadas. Específicamente, siendo X_{j-1} la muestra previa, X'_{j-1} la estimación de esta, X'_{j-2} la estimación de dos muestras anteriores, etc. En forma general un predictor lineal puede escribirse. ($X'_{j-1} = \hat{x}_{j-1}$)

$$g_j = \sum_{i=1}^k h_i \hat{x}_{j-i}$$

Los coeficientes h_i son los factores de ponderación y son escogidos para reducir los errores de estimación.

Por motivo de reducir el ruido de cuantización, se usa mayor frecuencia de muestreo. Empero hay otros ruidos; uno de los cuales se llama ruido de sobrecarga y se presenta cuando los niveles de cuantización $\pm k$ (en el caso delta, como salida de este modulador DPCM puede ser de varios niveles cuantizables) son muy pequeños para seguir a una señal que varía rápidamente. He aquí un compromiso, ya que el ruido de cuantización podría disminuir en gran parte reduciendo los niveles de cuantización (escalón $\pm k$) pero llegamos a incrementar el ruido de sobrecarga.

El siguiente gráfico explica este comportamiento tanto del ruido de cuantización como el de sobrecarga.



Para una onda sinusoidal tenemos que el valor es:

$$K \times f_s \geq 2 \times \pi \times f_m$$

f_s : frecuencia de muestreo

f_m : frecuencia de la muestra

Para otros casos, la experimentación es una herramienta decisiva; dependiendo de la frecuencia, se tiene que en la predicción influye directamente el k , como se nota en el diagrama anterior. El calcular el k óptimo es muy difícil con señales aleatorias, lo cual se lo hace experimentalmente y ensayando combinaciones lineales para formar ecuaciones de predicción.

Predicción isotrópica de intracuos para Y y color.

señal	frec	BW	BW	coef. de predicción			
	muest	horiz	vert	a1	a2	a3	a4
Y	10.125 MHz	5.0 MHz	4.5 MHz	1/2	0	1/4	1/4
R-Y	3.375 MHz	1.6 MHz	4.5 MHz	1/4	0	1/2	1/4
B-Y	3.375 MHz	1.6 MHz	4.5 MHz	1/4	0	1/2	1/4

datos obtenidos evaluando la ecuación

$$S_{intra} = a_1S'_1 + a_2S'_2 + a_3S'_3 + a_4S'_4$$

donde:

S'_i son las muestras obtenidas de un pixel cualquiera y S_{intra} es la predicción o comportamiento de un campo.

De esta manera, el método DPCM quien con cuya bondad se logra tener mayor información con pocos bits implica otros efectos a cuidar. El capítulo de DPCM es amplio, pero en el transcurso del desarrollo de los siguientes temas saldrán algunas de sus características.

2.2.- Codificación de una señal digital.-

Se ha dicho en forma general que una codificación es una DPCM; más cuanto esto se refiere a un tipo de modulación y no a un código específico.

En los esfuerzos por manejar el video de una manera más rápida, versátil y barata se fueron dirigiendo hacia un objetivo, "comprimir el video". El porqué de esta razón; es por la gran cantidad de datos que son manejados en la transmisión o para guardar información de imágenes digitales. Los datos son obtenidos por muestreo en una cantidad suficiente para determinar las escenas en niveles de brillo o dirección. Por ejemplo; para un tamaño común de una exploración de una escena de 512x512 elementos por campo y 8 bits por cada pixel, necesitaremos 2 Mbits por campo, esto implica 4 Mbits por cuadro.

El implementar esta cantidad de memoria en RAM, además de ser costosa, el manipuleo es complicado y limitado. Por ello comprimimos la imagen optimizando así todos los parámetros anteriores, permitiendo mayores bits para incrementar la definición en la imagen y control de errores.

Tipos de codificación.-

Los tipos de codificación son determinados por las diferentes formas de interpretar al video, variando las jerarquías de un parámetro a otro y por ello la forma de tratarle a la señal misma.

Hay dos siguientes

2.2.1.- ^{1.-} Compresión de imágenes.-

Al ser este tema una base para varias formas de codificación de imágenes la primera intención es dar una idea lo más subjetivo posible, para luego ir profundizando en todas sus características. Entre algunos parámetros que se debe cuidar con detenimiento es el ancho de banda, pues el diseñar filtros y circuitos con anchos de banda grandes a más de ser costosos son complejos en su diseño. En la transmisión no se debe exceder las normas establecidas para cada uso, así 4 MHz es para T.V.; si lo muestreamos con 8 bits podría requerirse un ancho de banda de 32 MHz cuando transmitimos usando un proyecto de modulación digital. Para ello los algoritmos serán los encargados de cuidar estos

parámetros para minimizarlos.

2.2.1.1.- IFS (Iterated Function Systems).-

La teoría de un IFS (sistemas de funciones iterativas) es una extensión de la geometría clásica de las transformaciones afines; esto define y transporta cuadros intrincados, por medio de cálculos iterativos (sucesivos) de ecuaciones pre-determinadas.

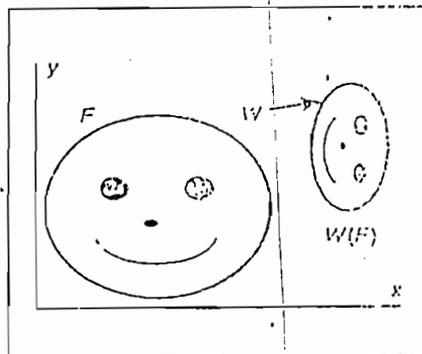
Una transformación afín puede ser descrita como combinación de rotación, escalamiento y traslación de ejes coordenados en n espacios dimensionales. Por ejemplo en una pantalla de T.V. en 2 dimensiones sería suficiente, así:

$$W(x,y) = (0.5x + 0.25y + 1 ; 0.25x + 0.5y + 2)$$

matricialmente

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0.5 & 0.25 \\ 0.25 & 0.5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

esta transformación mueve del punto (0;0) al punto (1;2) o del punto (-1;0.5) al punto (0.625;2) según su punto de evaluación; gráficamente se puede entender así:



La forma general de una Transformación Afin (TA) es:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}$$

$$= \begin{bmatrix} ax + by + e \\ cx + dy + f \end{bmatrix}$$

donde

$$\begin{aligned} a &= r \cdot \cos\theta & b &= -s \cdot \sin\phi \\ c &= r \cdot \sin\theta & d &= s \cdot \cos\phi \end{aligned}$$

r: factor de escala en x

s: factor de escala en y

θ : ángulo de rotación en x

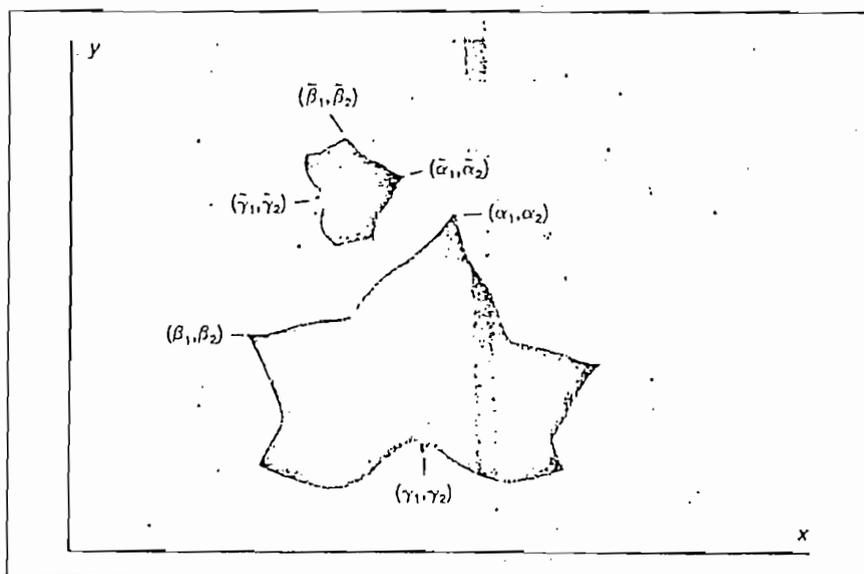
ϕ : ángulo de rotación en y

e: traslación en x

f: traslación en y

Los diferentes tipos de efectos que se consiguen por medio de la Transformación Afin, depende de los valores que tomen a, b, c, d, e, f. cuyos valores ya predeterminados e implementados circuitalmente o grabados en memorias de lectura son los que se selecciona en los equipos. Estos valores son encontrados con anterioridad resolviendo ecuaciones del siguiente tipo: (la letra gamma lo sustituiremos por su mayúscula Γ)

$$\begin{array}{ll} \text{sist} & \alpha_1 a + \alpha_2 b + e = \alpha'_1 & \alpha_1 c + \alpha_2 d + f = \alpha'_2 \\ \text{ecc} & \beta_1 a + \beta_2 b + e = \beta'_1 & \beta_1 c + \beta_2 d + f = \beta'_2 \\ (1) & \Gamma_1 a + \Gamma_2 b + e = \Gamma'_1 & (2) \Gamma_1 c + \Gamma_2 d + f = \Gamma'_2 \end{array}$$



Notese que la implementación será por resistencias que permitan esta compresión, además de operacionales que simulan los productos y convoluciones.

Empero un IFS es una colección de Transformaciones Afines de contracción (W_i); cada contracción tiene una probabilidad asociada p_i , determinando así la importancia o jerarquías de unas sobre otras.

Los diferentes escogitamientos de probabilidades en general no determina a las diferentes imágenes, pero sí influye en la definición de las imágenes o *segmentos* de ellas, que son exploradas. Así el tiempo de permanencia del explorador o scanning es

$$t_s = \text{area de } W_i / \text{area de } I$$

por segmento de exploración

Para una buena asignación de la probabilidad P_i se tiene:

$$p_i = (a_i * d_i - b_i * c_i) / \sum_{k=1}^n |a_k * d_k - b_k * c_k|$$

En un proceso de compresión y descompresión tenemos que una imagen entrante es rota en segmentos usando técnicas de procesamiento de imágenes, estas componentes de imágenes son recogidas en las librerías del IFS usando el teorema de Collage que dice: "Una 'imagen' puede ser dividida en áreas, buscando iterativamente coeficientes de los valores propios para las Transformaciones Afines, tal que al reconstruir la imagen coincidan de mejor manera todos los linderos de las componentes de imagen; esto no da facilidad de trabajo sin perder información del "Todo".

Cuando la imagen es reconstruida; los códigos IFS son ingresados al algoritmo iterativo aleatorio. (Anexo el programa). Es decir al cuadro entrante lo segmentamos tal que coincidan todos sus linderos de los segmentos asociados a W_m en las Transformaciones Afines contractivas. Podemos concluir que un todo se halla fragmentada en $W_1(s) \cup W_2(s) \cup W_3(s) \cup \dots$ y estos los hallamos resolviendo ecuaciones semejantes a sist ecc 1 y al sist ecc 2 antes descritas.

Por ello el grado de compresión depende del algoritmo usado; tanto en rapidez de ejecución como en su exactitud para limitar las fronteras de cada transformación afin contractiva y su número.

La segmentación puede hacerse simple o direccionada; estas últimas son conocidas también como técnicas de codificación de la segunda generación.

Un grupo de estas está caracterizado por el uso de operadores locales, los cuales son respuestas de pulsos de 2 dimensiones de bancos de filtros convolucionando los datos de imágenes en la sensación convencional. Cada respuesta de impulsos, está diseñada para extraer los rasgos de textura condicional o dependiente. Las salidas de estos filtros son luego combinados apropiadamente para obtener el mensaje que va a ser codificado. Codificación Piramidal y la predicción codificada no estacionaria anisotrópica son ejemplos más relevantes de este método.

Los métodos del segundo grupo usan acceso diferente; Estos intentan describir una imagen en términos de contorno y textura, que parecen ser más naturales coincidentes con el concepto de la psicología de la visión.

2.2.1.2. Q.- Codificación por segmentación.-

Aquí hay varias formas de desarrollar la segmentación, por ello se define y se examinan los segmentos; empezando de pixeles particulares, a sus pixeles adyacentes para seleccionar los que comparten las mismas propiedades. El proceso se detiene cuando son "etiquetados" con algún distintivo especial todos los pixeles conectados que comparten las mismas propiedades, produciendo una sola

región.

Otras regiones comienzan a aparecer desde el primer pixel que no contenga las mismas propiedades que el grupo contiguo, por ejemplo estas propiedades pueden ser los mismos niveles de gris.

Para evitar en lo posible regiones que no corresponden a superficies visuales de objetos reales; procesamos la imagen para quitar granularidades locales que afectan la definición del contorno, para ello usamos filtros de gradiente inversa y, a su vez hay 2 procedimientos utilizados para evitar este problema:

- Introduciendo alguna distorsión, eliminando regiones insignificantes y sus contornos.
- El uso de una mayor afinación de las propiedades que es igual que introducir complejidad no deseada.

Cada región puede ser caracterizada por un nivel medio de gris y localizada por sus puntos de contorno. En el siguiente paso, la diferencia entre la imagen original y este resultado intermedio es aproximado sobre cada región por un polinomio 2-D de los mínimos cuadrados, apto para introducir más realismo. El toque final es dado por un ruido equivalente a la granularidad removida durante pre-procesamiento.

La introducción de un mayor refinamiento generalmente nos encamina a dramáticos incrementos en complejidad de

simulación en computadora. Por ejemplo, una región creciente con un polinomio tipo 2-D, cuyos coeficientes son obtenidos por cada nuevo pixel incluido en la región, requieren varias horas de trabajo del CPU, y esto para pocas regiones. Por ello, para evitar esto usamos procesos llamados Split -and- Merge (corte y pegado). Para encontrar la segmentación correspondiente, se necesita una aproximación adaptiva para dar la naturaleza no estacionaria de las imágenes, esto es haciendo un análisis local y luego global de la imagen por modificaciones sucesivas de la forma de segmentación.

La decisión para segmentar una región será obtenida si esta está atravesada por un contorno (Según la experiencia, se requiere el contorno de una imagen para controlar la segmentación).

Los algoritmos usados para la extracción de contornos y por ello para su segmentación, estudiados en el artículo de " J.K. Canny "Finding edge and lines in images" M.I.T. Artificial intelligence Laboratory, Rep AI-Tr-720, 1983" necesitan de una aproximación de datos, así:

Si fijamos un conjunto vectorial G de funciones analíticas 2-D tipo $\Phi_i(x,y)$ ($i=1,\dots,r$)

$$g'(x,y) = \sum_{i=1}^r u_i \Phi_i(x,y)$$

pasando a notación vectorial

$$g'(x,y) = \Phi^T(x,y) \cdot u$$

Y si denotamos por k y l las coordenadas discretas de un plano de imagen, el conjunto de datos será:

$$g(k_j, l_j) \text{ con } j=1, \dots, N$$

El problema de aproximación es el escogitamiento de los coeficientes óptimos u_i .

Aplicando el concepto de medida de distancia: $g'_1(x,y)$ es mejor que $g'_2(x,y)$ en aproximación a $g(x,y)$ ssi

$$d(g, g'_1) < d(g, g'_2)$$

siendo d la distancia euclidiana

Como los contornos son preservados por la segmentación, la señal en cada región deberá ser armoniosamente ajustado al pixel original, encontrando el óptimo conjunto de parámetros u_i ; esto es equivalente a resolver el sistema de r ecuaciones lineales.

$$\partial (g, g') / \partial u_i = 0 \quad i=1, \dots, r$$

$$d(g, g') = \sum_{j=1}^n (e_j)^2 = (g - g')^T (g - g')$$

Los vectores g y g' contienen el valor original y aproximado respectivamente sobre el dominio D de cualquier 'imagen' a lo largo de un reticulado cartesiano tradicional, y e_j define el error en la localización de cada pixel (k_j, l_j) ($j=1, \dots, N$)

$$\text{Denotando matricialmente a } \text{---} \rightarrow u = S^{-1} Z^T g$$

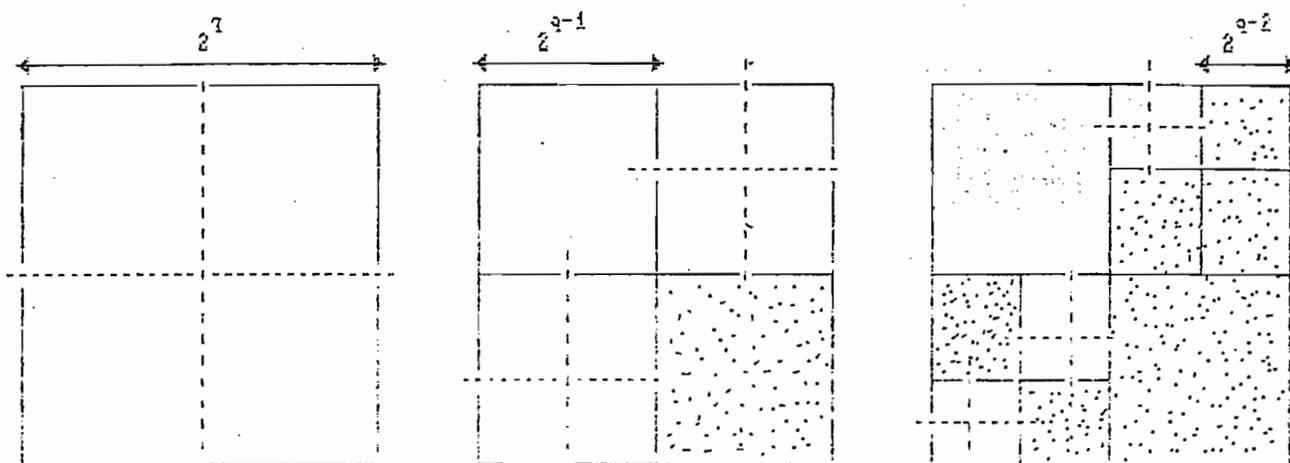
donde Z^T representa una matriz $(N \times r)$ de columnas de vectores $[\Phi(k_1, l_1), \Phi(k_2, l_2), \dots, \Phi(k_N, l_N)]$ y $S = Z^T Z$.

Se llega a la solución de este computo evaluando los vectores $\Phi_j(k_j, l_j)$. Se puede disminuir los cálculos

computacionales por medio de las aproximaciones de los mínimos cuadrados, y algoritmos de split-and-merge.

El algoritmo de Split es para la partición inicial de las regiones de una imagen, por procedimientos iterativos de subdivisión; este continúa hasta que se alcance un error pre-fijado, es decir cuando la región alcanza tamaños de un pixel o cuando una subdivisión puede causar que la matriz S sea singular, esto es que el número de puntos N es más pequeño que el número de funciones de aproximación (r).

Una región puede ser separada en más de 2 partes al mismo tiempo, siendo no aconsejable separar en muchas regiones, aunque computacionalmente se las puede hacer en regiones similares, por ejemplo en cuadros progresivos. Esto si se asume que la imagen es de dimension 2^a en cada dimensión. Empezando con una imagen de dimensiones $2^a \times 2^a$ la mejor aproximación para una imagen entrante de los mínimos cuadrados se evalúa con un conjunto de funciones polinomiales 2-D. Si la aproximación es satisfactoria el proceso se detiene ahí; de lo contrario la imagen es dividida en 4 cuadros idénticos de tamaño $2^{a-1} \times 2^{a-1}$ y el proceso es similar e iterativo hasta que la calidad sea aceptable.



cuadros con calidad de aproximación satisfactoria



forma de partición en el siguiente nivel.

Para tener un tiempo límite en la ejecución, el proceso se detiene, independientemente de su aproximación cuando $2^{2(q-1-1)}$ es más pequeño que r , definiendo luego a l como el último nivel de subdivisión. Experimentalmente se encontró que este nivel tiene una buena aproximación, dado por los valores pequeños que toma r .

Según los análisis de errores, los datos así obtenidos son procesados en los siguientes pasos.

Cada término de la matriz S puede ser expresado como:

$$S_{m,n} = \sum_{j=1}^n \Phi_m(K_j, l_j) \Phi_n(K_j, l_j) \dots$$

cada coordenada k_j o l_j tiene $\sqrt{(N)}$ veces el mismo valor si tiene n puntos en el cuadro.

Asumiendo que:

$$\Phi_m(x,y) = x^a y^b \quad \text{y} \quad \Phi_n(x,y) = x^c y^d$$

se tiene que:

$$S_{m,n} = \sum_{n=1}^{\sqrt{N}} n^{a+c} \sum_{n=1}^{\sqrt{N}} n^{b+d}$$

Y realizando las operaciones respectivas que será una multiplicación de series de potencias, podemos hacer ya los computos de las areas S, pero por ser regiones cuadradas una solución más fácil es separando la función de aproximación $g'(x,y)$

$$g'(x,y) = \sum_{i=1}^{r_1} \sum_{j=1}^{r_2} (u_{ij} x^i y^j)$$

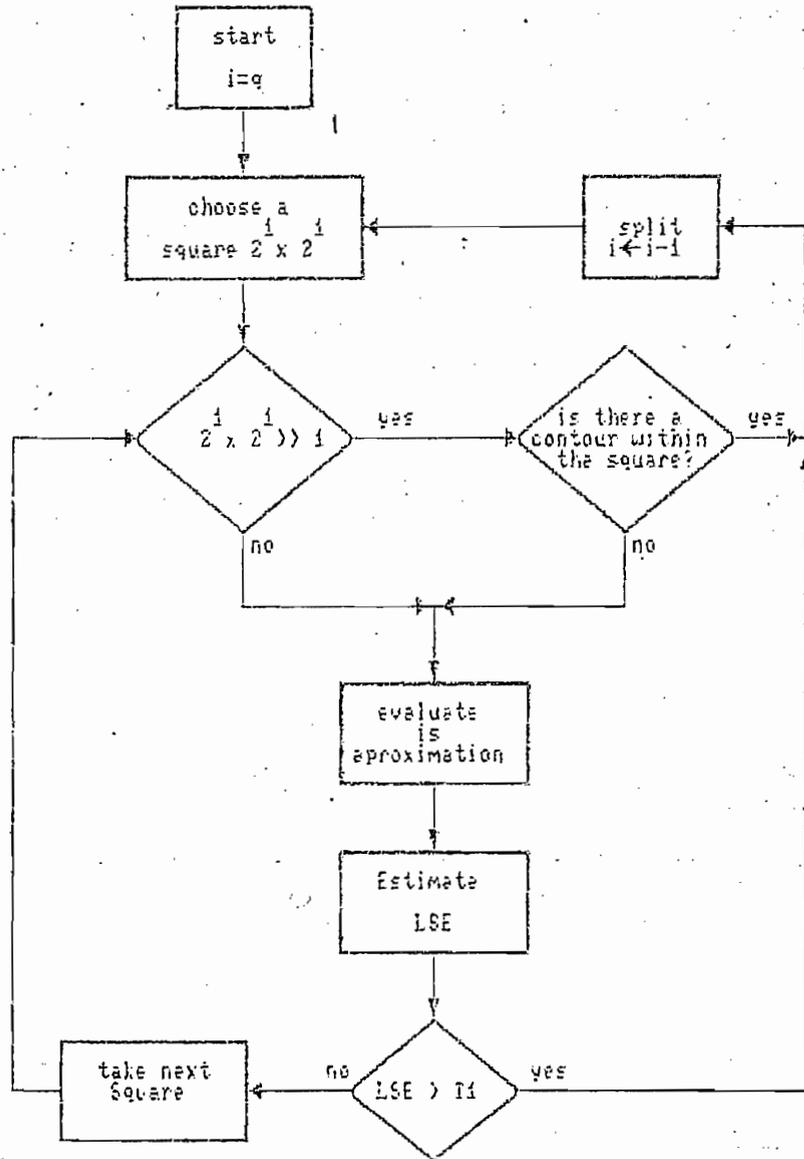
y computando en notación matricial a los coeficientes u_{ij} de aproximación por medio de los minimos cuadrados:

$$U = (V_K^T V_K)^{-1} V_K^T G V_1 (V_1^T V_1)^{-1}$$

Nota.- referirse para esta demostración de las matrices a los papeles: [1] M. Eden, M. Unser, and R. Leonardi, "Polynomial representation of picture". Signal Processing, vol 10, nO 4 pp 386-393, 1986. y [2] R. Leonardi, "Segmentation adaptive pour analyse et codage d'images a haute performance," PhD thesis, Dept Elec. Eng. Swiss Federal Institute of technology, Lausanne, Switzerland 1987.

Según datos obtenidos por procedimientos de algoritmos matemáticos de matrices y computos se obtiene un flujograma

del proceso, mostrado a continuación:



Número de operaciones requeridas para el computo de coeficientes de aproximación.

size	N=4 ²	N=8 ²	N=16 ²	N=32 ²	N=64 ²
Poly					
r ₁ =2 r ₂ =2	64 80	176 272	592 1040	2192 4112	8464 16400
r ₁ =2 r ₂ =3	86 132	206 420	638 1572	2270 6180	8660 24612
r ₁ =2 r ₂ =4	112 192	240 576	688 2112	2352 8256	8752 32832
r ₁ =3 r ₂ =3	138 225	318 657	966 2385	3414 9297	12918 36945
r ₁ =3 r ₂ =4	180 336	372 912	1044 3216	3540 12432	13140 49296

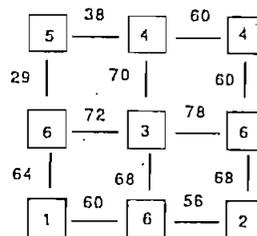
El algoritmo de Merge se puede decir que es el proceso inverso del split, en donde algunos cuadrados pueden contener contornos mientras otros definen partes de areas homogéneas, estos últimos nos servirán como semilleros para el proceso de merge, los pixels que contienen contornos tendrán al mismo criterio de homogeneidad.

Similarmente puede ser medido con respecto al grado de modificación de las funciones de aproximación $g(x,y)$ sobre 2 regiones. El problema es fijar el propio umbral para cada uno. La idea básica es mirar las regiones adyacentes pares que inducirán, la menor degradación. Para desarrollar esto se construye una gráfica de regiones adyacentes (RAG); cada nodo en un RAG representa una región y cada lazo define al adyacente. Luego se asigna un valor

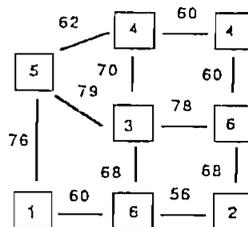
de ponderación a cada lazo para indicar el grado de no similitud entre nodos, es decir, 2 regiones. En cada pazo el lazo con menor diferencia ("costo de degradación") identifica el merge de 2 regiones similares más cercanas. El RAG es computado con una envolvente de segmentación; el nodo resultante de la asociación de 2 regiones se enlaza para la unión de todos los otros nodos conectados a 2 regiones. Para dar una medida similar de una función de costo de degradación a un conjunto de funciones de aproximación, se asegura un mínimo para ponderar, la cual será la partición óptima de imagen, excepto en imágenes patológicas, ejemplo:

6	3	8	5	5	8
6	4	1	2	2	1
8	7	8	3	8	7
5	2	1	0	7	2
2	1	4	9	5	1
1	0	5	4	0	2

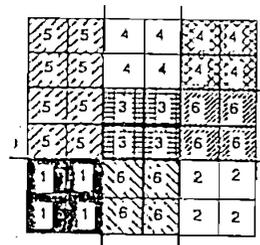
a) Original image
(grey levels)



b) Initial Segmentation



c) Segmentation after
one step of merging

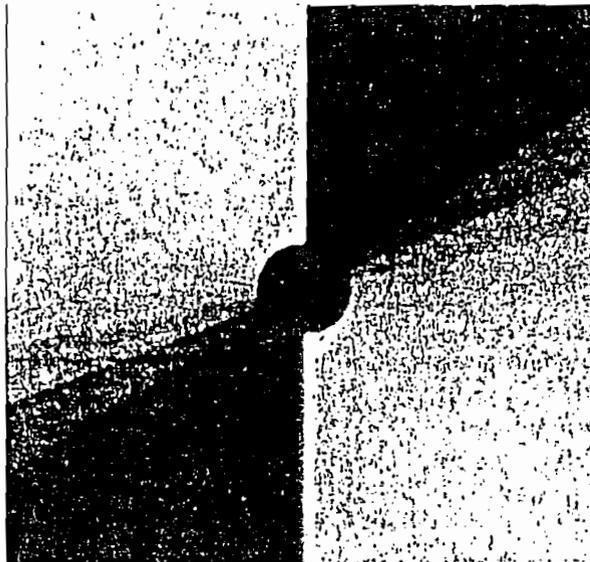


d) Reconstructed Image

Más cuanto esto no es la única forma de comprimir una imagen; también hay otros que usan implementación circuital como base, así tenemos la de descomposición direccional.

2.2.1.3.- ^{3.} Codificación por Descomposición Direccional.-

Este es implementado con bancos de filtros de 2-D, produciendo una componente de imagen pasa bajos y N direcciones de imagen, conteniendo componentes de alta frecuencia en direcciones dadas entre 0° y 180° cuantizadas en un pequeño N (8 a 16) de tal manera que cada filtro direccional cubra cada sector del dominio de Fourier.



Esto se puede ver mejor con un cuadro de imagen desmenuzado direccionalmente, donde se nota las ortogonalidades de los perfiles (ver anexo 1)

En el gráfico note que cada dirección de imagen se filtra en pasa altos y los pasa bajos es la componente de imagen perpendicular.

Las componentes de baja frecuencia son codificadas por submuestreo y cuantización o por "transformación codificando" usando técnicas de la primera generación.

Las componentes de alta frecuencia son usadas para detectar flancos (contornos). Empero la posición y magnitud también son codificadas. La longitud de "corrida de exploración" es usada para codificar la posición, y la magnitud es codificada por la fijación o determinación de longitud de las palabras código.

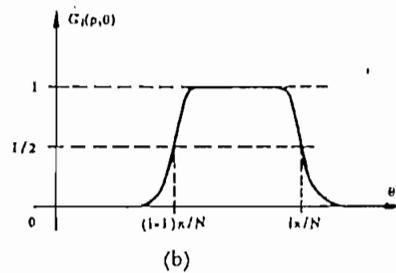
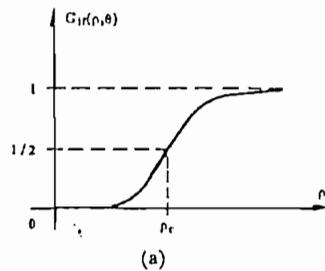
Estas son implementadas con interpolación y re-muestreo; en la reconstrucción de imágenes direccionadas, se invierte el proceso para obtener la direcciones originales, usando el mismo procedimiento. Así la decodificación es llevada a cabo como sigue: las componentes bajas de frecuencia son submuestreadas. Las direcciones en las imágenes ya direccionadas (componentes) son reconstruidas usando sintetizadores de perfiles de cruces por cero, en los cuales los parámetros son emparejados para la decodificación de posición y magnitud.

Así se tiene que el comportamiento de los filtros serán:

$$a) \quad G_{1r}(f, g) = \begin{cases} 1 & (f^2 + g^2) < f_c^2 \\ 0 & \text{otros} \end{cases}$$

$$b) \quad G_1(f, g) = \begin{cases} 1 & (f^2 + g^2) \geq f_c^2 \\ & (i-1)\pi/n < \text{tg}^{-1}(g/f) \leq i\pi/n \\ 0 & \text{otros} \end{cases}$$

con $i = 1, \dots, N$



donde:

$$G_{1r}(f, g) = H(r, [r_c, \sigma_r])$$

$$G_1(f, g) = [1 - H(r, [r_c, \sigma_r])] [1 - H(\theta, [i\pi/N, \sigma_\theta])]$$

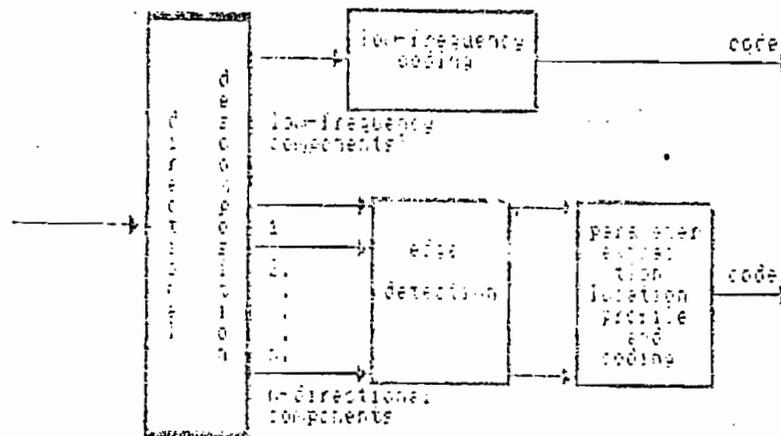
Donde σ_r y σ_θ determinan la desviación estándar de la ventana gaussiana en dirección principal y perpendicular respectivamente; σ_r actúa en vibración u ondulación perpendicular a la dirección del flanco, σ_θ reduce la dispersión en la dirección del flanco.

Después de la descomposición direccional, cada componente es codificada independientemente. Por definición, las imágenes de baja frecuencia es una señal de

banda, limitada por f_c ciclos por grado en ambos direcciones. Pudiendose usar dos métodos equivalentes para codificar esta señal: "transformar codificando" o submuestreo en el dominio de la imagen.

La experiencia indicada muestra que la de submuestreo es ligeramente mejor que el "transformar codificando", ya que da la misma cantidad de bits usada para codificar componentes de baja frecuencia, sin embargo parecen ser mejores en la calidad los cuadros reconstruidos obtenidos por submuestreo. Como quiera la razón de compresión más allá de 5 a 1 para esta imagen pasa. Bajos no son alcanzados por estas técnicas. Para ello se requerirán otro pos-procesamiento para compresiones más allá de 10 : 1.

La estrategia de codificación debe ser mejorada en la detección de flancos, representación y codificación.



El bloque de Detección de flanco que es una parte importante y amplia ante todo nos dará la exactitud de la detección y por ello mejoraría en la definición.

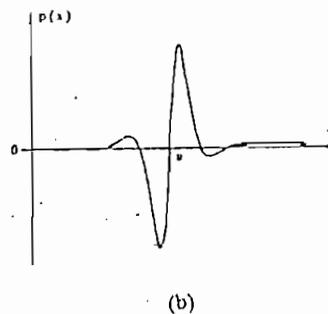
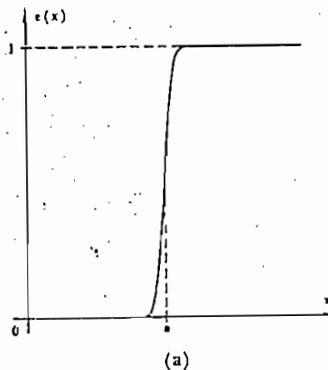
Existen dos maneras de utilizar estas herramientas .

La primera es detectar los flancos con cada imagen direccionada, la segunda es detectar en el campo isotrópico, es decir las componentes de alta frecuencia usando la imagen direccional.

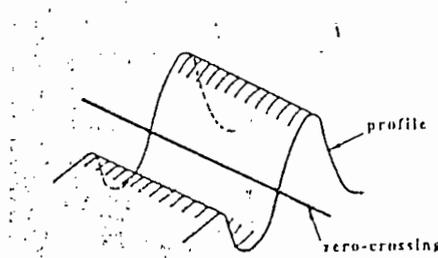
La detección de cruce por cero es el primer paso en la detección de flancos, por ello el estudio de la respuesta de alta frecuencia de los filtros, como los hay en un flanco o función escalón. Si el filtro es ideal con una agudeza de transición en su frecuencia de corte; la respuesta a un flanco ideal tiene un cruce por cero en la localidad de un flanco y a los lados de los lóbulos laterales. Los lóbulos laterales son particularmente distribuidos en la localización de cruce por cero e introducen ambigüedades. Para eliminar esta, la respuesta del impulso de longitud infinita necesita ser truncada por una función ventana. La transformada de Fourier de esta función convoluciona la respuesta ideal y dirige a la respuesta de frecuencia de un filtro suavizador o aplanador. La respuesta de este filtro para el mismo flanco ideal es suavizada, donde solamente el cruce por cero central está presente.

Luego se desarrolla una detección no ambigua de este cruce por cero separando fácilmente las partes positivas y negativas de la señal.

Para el caso de un flanco real y no ideal, como se muestra en la figura se tiene:



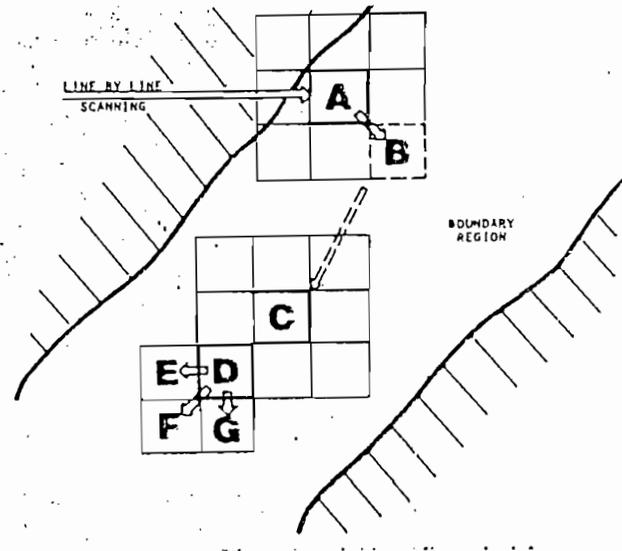
El signo de la pendiente del cruce por cero puede ser falsa, según la detección sea en $a + \epsilon$ del punto a del flanco. Para mejorar este desarrollo, se toma en cuenta una nueva medida basada en la energía del perfil local.



De una imagen direccionada, el perfil es extraído sobre una línea perpendicular a la dirección del flanco y su energía es computada como la suma de los cuadrados de las muestras a lo largo de esta línea. La extensión de el perfil es determinado por el primer lóbulo. Si la componente isotrópico de alta frecuencia es usada (la suma de imagen direccional) la energía es computada a lo largo de todos los posibles perfiles direccionados y luego promediados. Es así como la importancia de un cruce por cero radica en ser proporcional a la medida de la energía.

Consecuente con esto, se aplica una técnica más refinada, conocida como ridge-riding (cabalgar haciendo lomos) o alineación de flancos.

En la alineación de flancos, se usa un primer umbral, llamado Umbral de Contorno Estelar (CST), para detectar puntos de flancos correspondientes a largas magnitudes. Luego, empezando por este punto, se traza puntos de flancos con ventanas 3x3 en la dirección de mayor magnitud. El trazo continúa hasta un segundo umbral, llamado Umbral de Punto de Flanco (EPT)

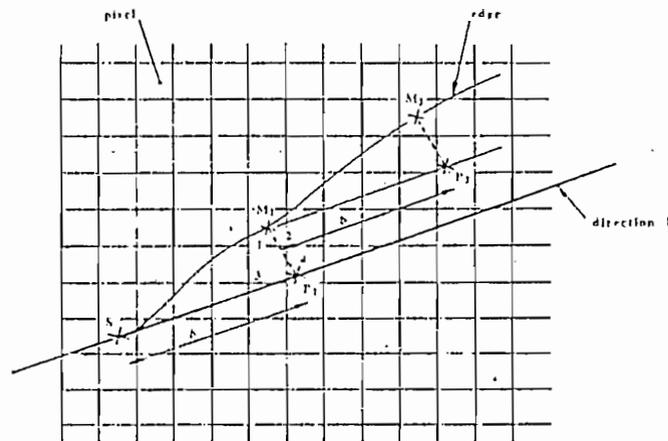


Submuestreo de Flancos y Codificación

Empecemos definiendo a S como un punto de flanco en la dirección de imagen número k y Δ la razón de submuestreo, denotemos a P_1 el punto tal que $SP_1 = \Delta$ a lo largo de la dirección $k \cdot n/N$. Notemos que P_1 está localizado en un punto cuyas coordenadas son reales. para luego tener valores enteros. Es distinto que este punto caiga exactamente sobre la grilla cartesiana de digitalización. M_1 es el flanco más cercano al punto P_1 ; en efecto M_1 será un pixel (en la grilla discreta) en las vecindades de P_1 , y representaremos por d_1 entre 1 y 4 dependiendo del pixel asignado a M_1 . Empezando desde M_1 y en la misma dirección, el procedimiento se repite hasta obtener una distancia d_2 . El proceso se detiene cuando la parte permanente de el flanco

tiene una longitud menor que $\Delta/2$.

Con esta representación y usando un código Huffman, se tiene que para un flanco de n muestras, 13 bits se requieren para empezar a marcar el punto S , 2.5 bits por n , y 2.6 por cada d_1 dandonos un total de $15.5 + 2.6n$ (bits). El código Huffman estadístico y estimado sobre 16 direcciones de imágenes da resultado y es particularmente interesante para flancos largos, para lo cual es suficiente 1 bit/contorno para que el punto sea fácilmente obtenido; para un $\Delta = 5.36$ (bits) se codifica un flanco teniendo 40 puntos. El valor de Δ es escogido de acuerdo a la razón de compresión y al flanco más pequeño que pueda no ser eliminado.



Este tipo de descomposición es en base a filtros; ya detectado el flanco, los datos guardados de cada dirección

serán tratados por el algoritmo implementado tanto en software o hardware con memorias EPROM y microprocesadores, lo cual lo hacen en las tomografías, pero aquí como es de suma importancia el determinar la posición exacta de cada punto, es decir, determinación de cada una de las tres dimensiones largo ancho y profundidad, para ello, se usa más de un microprocesador para poder computar por triangulación esta posición.

Los siguientes tipos de codificación nos permiten alcanzar altos grados de compresión y con un bajo nivel de distorsión. Uno de estos es usar la técnica de Vector de Cuantización.

U.7

2.2.2.1.- Codificación de imágenes usando Vectores de Cuantización.-

Esta técnica usualmente utiliza lo psicovisual como redundancia estadística en los datos de imágenes para reducir la razón de bits. Una deficiencia con todas estas técnicas convencionales de codificación es que la cuantización es desarrollada sobre un muestreo de evaluación real e individual de formas de onda o pixels de imágenes. Por ejemplo: el "codificar transformando" hace esto, primero bloques de transformación de un vector y luego cuantización escalar, de las muestras transformadas en el vector.

Definición de un Vector de Cuantización.-

Un vector de cuantización puede ser definido como un mapa Q de K dimensiones de espacio euclidiano R^K dentro de un subconjunto finito Y de R^K . Así:

$$Q: R^K \rightarrow Y$$

Donde $Y = \{x_i; i=1, 2, \dots, N\}$ es el conjunto de vectores de reproducción y N el número de vectores en Y . Esto puede ser visto también como una combinación de dos funciones; un encoder, quien mira las entradas de los vectores x y un generador de direcciones de los vectores de reproducción especificadas por $Q(x)$, el decodificador el cual usa las direcciones para generar los vectores de reproducción \hat{x} , luego con el mejor mapa Q se minimiza $d(x, \hat{x})$.

Una medida de distorsión por codificación de forma de onda es el error cuadrático de distorsión dado por:

$$d(x, \hat{x}) = \|x - \hat{x}\|^2 = \sum_{j=0}^{K-1} (x_j - \hat{x}_j)^2$$

Se puede usar también un medio cuadrático ponderado (WMSE) como medida de distorsión.

Diseñar un vector cuantizador es obtener N vectores de reproducción, tal que minimize la distorsión esperada.

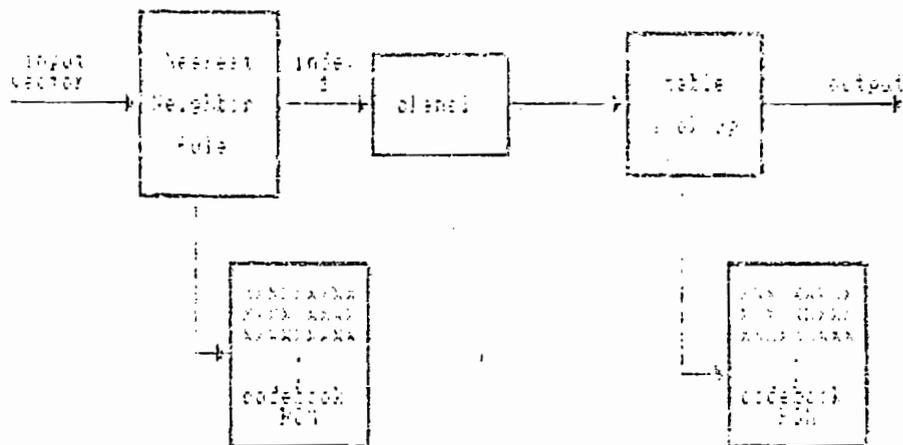
La distorsión esperada es aproximada por el tiempo promedio del error cuadrático dado por:

$$D(x, q(x)) = (1/N) \sum_{i=0}^{N-1} d(x_i, \hat{x}_i)$$

Existen muchas técnicas para obtener el codebook inicial. Linde (investigador sobre procesamientos de señales como video) usa una técnica particular donde el

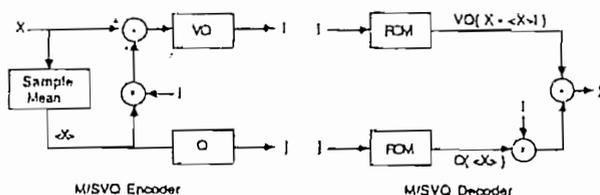
centroide, es calculado y separado en 2 vectores cerrados según la secuencia de instrucciones. Cada vector resultante puede ser después partido en 2 vectores y el procedimiento siguiente es repetido hasta N veces desde que se crea el vector reproductor.

En el proceso de diseñar un codebook llegamos a vectores de cuantización espacial que pueden ser adaptivos y no adaptivos y han sido diseñados para codificar cuadros de imágenes. El concepto básico envuelve el particionamiento de una imagen dentro de vectores del tipo 2-D. Cada vector luego es comparado por un conjunto de vectores standar guardados en una ROM, y transmitida una palabra código identificando la mejor pareja. El receptor reconstruye la imagen usando el correspondiente modelo en lugar del vector original.

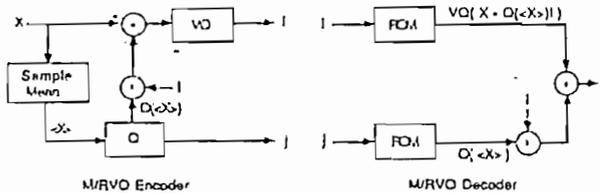


Este diagrama muestra un simple sistema de codificación de vector de cuantización.

Otra técnica usada es la llamada Mean/Shape VQ. El muestreo medio es escalarmente cuantizado y el vector de error resultante obtenido por sustracción del muestreo medio de un vector entrante.



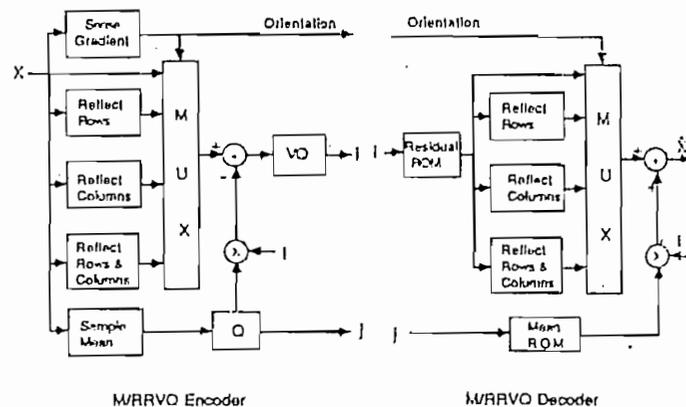
Una variación de M/SVQ fue también introducido y es llamado Mean/Residual VQ.



En este sistema de codificación, el muestreo medio es primero escalarmente cuantizado y luego sustraído del vector, tal que cualquier error en la cuantización de la media es incorporado en el vector de error.

El sistema M/RVQ reduce la distorsión de bloque, que es causado por gruesas cuantizaciones del muestreo medio en el sistema M/SVQ.

Algunos vectores cuadráticos difieren solamente al ser reflejados a sus ejes vertical y horizontal; así también es posible transmitir esta orientación haciendo corresponder vectores de codebook.

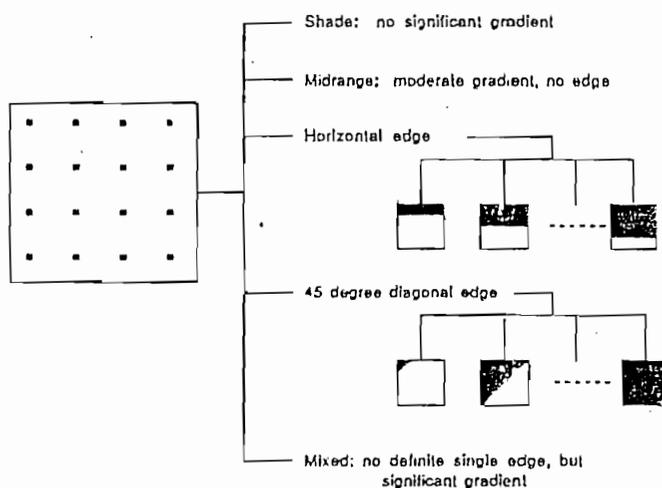


Esto es llamado Mean/Reflected VQ. Las mejoras de esta aproximación es que, enviando la correspondiente reflexión de información no se tiene que guardar todos los posibles vectores código, aunque se tiene que aumentar 1 bit para enviar la información acerca de la simetría del vector bloque.

Para clasificar VQ, los vectores codificados de imagen en una baja cantidad de bits, usualmente padecen de distorsión de flancos y estos no pueden ser reproducidos

perfectamente por un tamaño pequeño de código de libro. Normalmente se tiene una simple clasificación de las instrucciones de vectores en flancos y tipos de sombras seguido por un codebook separado, diseñado para cada clase.

959



Como es de gran importancia preservar la fidelidad de flancos en localización y orientación angular, cada bloque de flancos fueron subdivididos en más clases dependiendo de la orientación y localidad de los flancos.

Como así lo indica la figura anterior para la clasificación usada.

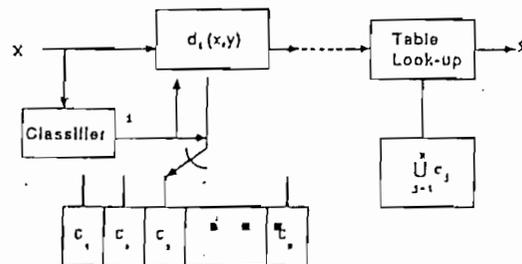
Los subcodebook fueron diseñados separadamente por cada clase usando el algoritmo LBG (Joseph Linde, Andrés Buzo, Roberto Gray, "An Algorithm for Vector Quantizer

Desing", IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. COM-28, No. 1 January 1980.).

Por ejemplo, C representa el overall del codebook y C_i representa el codebook para cada clase y también $C = \bigcup_{i=1}^M C_i$ asumiendo que hay M clases.

Los codebook fueron diseñados usando el algoritmo LBG para cada clase permitiendo la secuencia de instrucciones x_j [$j=1, \dots, n$] que serán clasificados en la apropiada clase x_j^i [$j=1, \dots, d$ $i=1, \dots, M$].

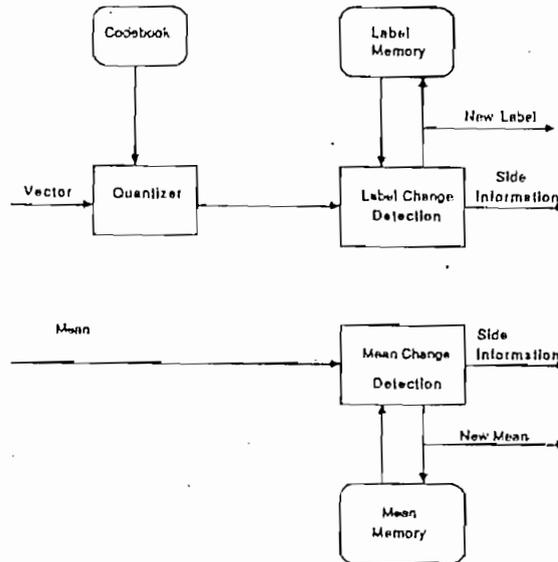
El algoritmo LBG es usado para minimizar la distorsión de clase $d_i(X, Y)$ donde X y Y son restringidos a la correspondiente secuencia de instrucciones x^i y el codebook C_i .



Se emplean operadores de gradientes de flancos en el mejoramiento de los flancos para la clasificación de cada vector de imagen. El llenado del codebook son con los VQ, tanto para codificación B/W como para cuadros de color.

En este sistema, adaptivamente fue basado sobre una

computación y transmisión de un pequeño codebook que empareja la estadística local de una imagen para ser codificada.



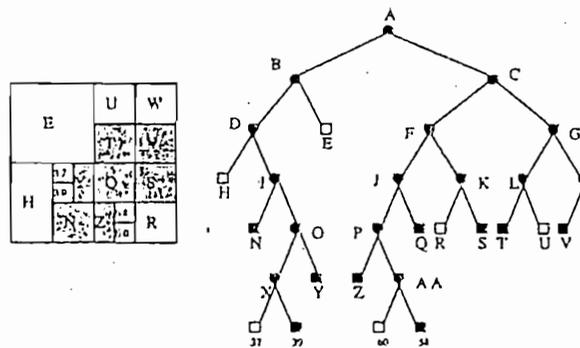
La imagen es subdividida en subimágenes no traslapadas, y para cada subimagen, un codebook consistente en cerca de 16 a 64 vectores representativos. Por cada subimagen se transmite la localidad generada por los vectores representativos del codebook, seguidos por la palabra código o etiquetados por los vectores de la imagen.

Las imágenes codificadas por esta técnica producen visualmente cuadros aceptables cayendo entre 1 y 1.5 bit-pixel.

En todo sistema existe elementos más jerárquicos que otros, así también en los VQ, un punto importante es que en

el sistema adaptivo SVQ, el tamaño de bloque es constante en todo el proceso.

Con esta técnica de jerarquización adaptivos de VQ, primero se usa un algoritmo de árbol cuadrático para la partición de imágenes dentro del bloque de dimensión 2x2, 4x4, 8x8 y 16x16. Esta información de partición de la imagen representada por un árbol cuadrático es contabilizado como una parte de la información para ser transmitida.



Los usos y variaciones que provee un VQ son grandes y permisibles de profundizar en estudios de la Teoría de la Información aplicados en ellos; una de estas variaciones es la codificación por transformación adaptiva en donde se diseña cada bit asignado matricialmente.

Cada sub-bloque de la imagen es clasificado en una de varias clases acorde a la actividad del contenido del bloque, y codificado por el bit apropiado asignado

incremento en la razón de bits, este método es mostrado en la siguiente figura:

```

región de
zona
codificada
7 5 4 3 3 2 2 2 1 1 2 1 1 1 1 1
5 6 2 2 2 1 0 0 0 0 0 0 0 0 0 0
4 2 1 1 1 0 0 0 0 0 0 0 0 0 0 0
4 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
3 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2 0 0 0 0 región de 0 0 0 0 0 0
3 0 0 0 0 vector 0 0 0 0 0 0 0
2 0 0 0 0 cuantización 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2 0 0 0 0 0 0 0 0 0 0 0 0 0 0

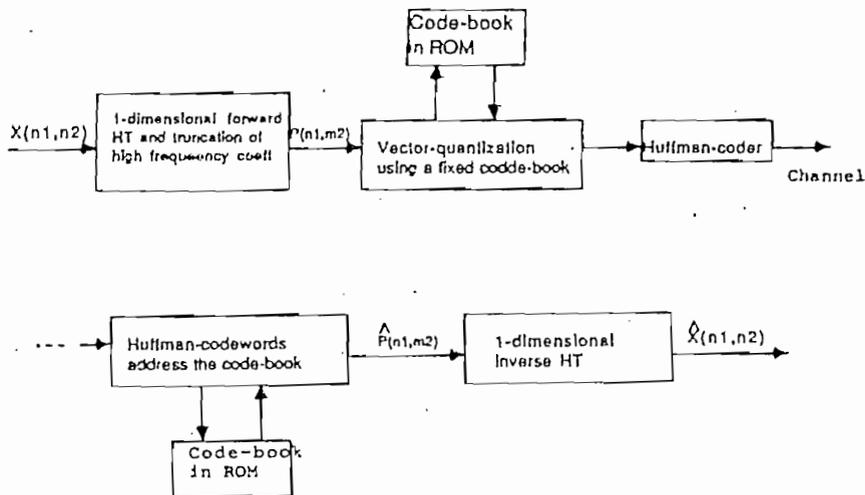
```

Es ahora como se puede ver dos partes; una de coeficientes de baja frecuencia codificados por un cuantizador escalar, y otro de coeficientes de alta frecuencia por un vector de cuantización. Esto nos muestra que las componentes de alta frecuencia pueden ser transmitidos con un pequeño incremento en la razón de bits, y la reconstrucción de imágenes es de mejor calidad con una baja razón de bits.

Cuantizador de vectores transformados de una-dimensión.-

En codificadores de una dimensión, las líneas de imágenes son transformadas por una transformación unitaria uni-dimensional y una matriz de asignación de bit que es usada para el código de coeficientes normalizados en cada

línea transformada. El cuantizador es un escalar donde la correlación entre los coeficientes en cada línea o previo a cada línea, no son incorporados en el cuantizador. La correlación con los coeficientes transformados en cada línea de imagen no es significativa desde que la transformación lineal es confiada a la decorrelación de muestreo con cada línea. Pero la correlación de interlínea es bien significativa y este puede ser aprovechada por ejecución de un VQ. Una de estas técnicas (usada por King y Nasrabadi) se muestra en el siguiente diagrama:



Entre algunas variaciones que sufre este tipo de codificación dependiendo de la aplicación que se lo vaya a tener, es decir, sus limitaciones en cuanto a definición, sensibilidad a ruido, y uso al cual se va a someter, como en la aplicaciones de video T.V., análisis de datos

telemétricos y de exploración médica.

Con el propósito de indicar la mayor cantidad de tipos de codificación o en su defecto las variaciones más sensibles de un tipo, que tienen mejor calidad en video veremos el siguiente tema.

2.2.3.- 5.- Codificación de imágenes basadas en transformación Ortogonal traslapada (LOT).-

"Transformar codificando" es reconocida como una de los métodos para la compresión de datos de imagen codificada. En este sistema la señal de video digital es típicamente dividido en bloques cada uno contiene 8×8 pixels, luego son sujetas a preservación de energía por transformación unitaria. El objetivo de esto es convertir estadísticamente elementos de cuadros dependientes (pixels) en un conjunto de coeficientes independientes esenciales de transformación, preferible empaquetando señales de energía o información en un mínimo número de coeficientes. El resultado de los coeficientes de transformación son cuantizados, codificados y transmitidos. En el receptor la señal de video es recuperada por computación de la transformación inversa, luego decodificando los datos transmitidos.

Si la señal F representa la imagen digitalizada la cual puede ser vista como una matriz de dimensiones $R \times R$, donde R es la resolución de la imagen, la representación de

la señal de video en el dominio de la transformación es la matriz F_1 comprendiendo a $R \times R$ como los coeficientes reales de transformación. Con una separación de transformación de 2-D, la matriz F_1 es obtenida de:

$$F_1 = T F T^t$$

donde T^t nos indica la matriz transpuesta. La matriz $T_{R \times R}$ es unitaria y representa la transformación kernel de una dimensión. Las filas de la matriz T de transformación son definidas como las funciones básicas de transformación. En los sistemas que transforman codificando, la prioridad para transformar la señal de video es sujeta a una segmentación de bloque en la cual la imagen codificada F de dimensiones $R \times R$ es dividida en subimágenes o bloques de datos de dimensión $N \times N$ pixels donde $R = mN$ tal que resultan subimágenes de $m \times m$. La transformación es aplicada para cada bloque independientemente. La segmentación del bloque puede ser representada por la siguiente estructura de bloque diagonal de una transformación kernel T unidimensional.

$$T = \begin{bmatrix} A & & & 0 \\ & A & & \\ & & & \\ & & & A \\ & & & & \dots & A \end{bmatrix}$$

Donde A se define como el bloque unidimensional de transformación kernel de $N \times N$. La matriz A debe asegurar que la matriz T sea unitaria, por ello A tiene que ser unitaria; en otras palabras, A es una pequeña matriz de transformación y tomando la transformación de la imagen F

es equivalente a tomar la transformación de dimensión $N \times N$ para cada bloque de F . Las N funciones de transformación básicas a_i , $i=1 \dots N$, son las filas de A

$$A = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix}$$

la "codificación transformando", aprovecha las mejoras de la correlación entre pixels adyacentes para reducir la redundancia. Porque de la segmentación de bloques, las dependencias estadísticas más allá de los linderos del bloque no son tomadas en cuenta dentro de esta consideración. Adicional a esto, con la razón de datos de aproximadamente 1 bit por pixel los bloques periféricos vienen a ser visibles. La visibilidad de los linderos puede ser reducida sobre-traslapando los bloques antes de transformar codificando.

Para mayor claridad de este tipo de codificación daremos un ejemplo y su definición formal.

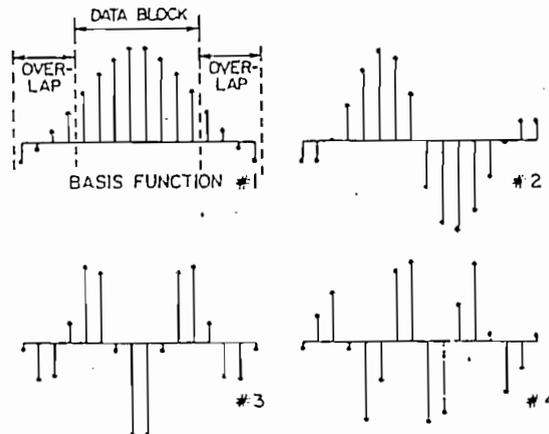
Suponga que f es una columna vectorial de dimensión R representando a una columna de pixels extraída de la imagen F . De la transformación unidimensional de f se obtiene f' , compuesta de coeficientes obtenidos como sigue:

$$f' = Tf.$$

Para cualquier operador T de transformadores de traslape o no traslape, la señal transformada f' puede dividirse en $m=R/N$ no sobretraslapada y m bloques independiente de N

muestras cada uno. La segmentación de bloque puede ser explicada por la siguiente propiedad de T : la función base de T , sobre la cual la proyección de f obtienen 2 bloques adyacentes de datos transformados de f , son idénticos después de un intercambio lineal de N muestras, ignorando efectos de campo. Está claramente ilustrada por la estructura de bloque diagonal de T , para el caso de no-sobretraslapada.

Se define una transformación ortogonal de traslape como una transformación unitaria separable, por la cual las funciones bases correspondientes a los bloques de datos adyacentes sobre-traslapan en el dominio de la imagen, por ejemplo si tenemos cuatro funciones básicas para el caso de dimensión del bloque $N=8$ y sobre-traslape $L=8$.



consecuentemente, una transformación ortogonal de traslape es definida por la siguiente estructura de bloques de sobre traslapación de sus transformaciones unidimensionales de

obtiene una no redundancia en la representación de la imagen digitalizada.

La matriz A es compuesta de N filas de vectores de funciones base a_i . Las funciones base de A para $L=N$ pueden ser escritas así:

$$a_i = (x_i, y_i) \quad i=1, \dots, N$$

donde x_i y y_i son dos filas de vectores de dimensión N , respectivamente, representando el primero y el último elemento de la función base a_i . Para asegurar que la T sea unitaria, la fila de vectores de T debería ser un conjunto ortogonal y ser normalizada. Dando la estructura de bloque de T , la condición de ortogonalidad produce los siguientes conjuntos de obligaciones sobre las funciones base:

$$(x_i)(y_j^t) = 0 \quad i, j = 1 \dots N$$

$$(x_i x_j^t) + (y_i y_j^t) = 0 \quad \text{para } i, j = 1 \dots N \text{ y } i \neq j$$

y según la normalización se tiene

$$(a_i)(a_i^t) = 1$$

Esto es lo que se puede decir acerca de la codificación ortogonal, ya que indicar con más detalle, nos llevaría a profundizar en las condiciones matemáticas, y por lo tanto sobre la teoría de la información codificada.

Por dar un bosquejo de otro tipo de codificación también ampliamente difundido, veremos.

2.2.4.- Otros tipos de codificación.-

2.2.4.1.- Pirámide Laplaciana como un código compacto de imagen.-

La característica común de las imágenes es que los pixels con vecindades tienen alta correlación. Pero representar la imagen directamente en términos de valores de pixels es ineficiente, mucha de la información es redundante; por ello el primer paso en el diseño de un código eficiente y comprimido es encontrar una representación en la cual decorrelaciona los pixels.

En un código predictivo, los pixels son codificados secuencialmente en formato rastreador. Para codificar cada pixel, su valor es predicho por la codificación previa de los pixels y siguiendo el rastreo de línea. El valor predicho de los pixels, los cuales representan la información redundante, es sustraída del valor actual del pixel, y solamente la diferencia, o el error de predicción es codificado, este procedimiento es llamado causal. Las restricciones para la predicción causal facilita la decodificación: para codificar un pixel dado, su valor predicho es recomputado de los pixels ya decodificados en las vecindades, y sumados para guardar la predicción de error.

Predicción no causal, basada en una vecindad simétrica centrada en cada pixel, debería producir más exactitud en la predicción, y gran compresión de datos.

Si se define la imagen original como $g_0(ij)$, y a $g_1(ij)$ el resultado de aplicar un filtro pasa-bajos a $g_0(ij)$. El error de predicción es L_0 y es:

$$L_0(ij) = g_0(ij) - g_1(ij).$$

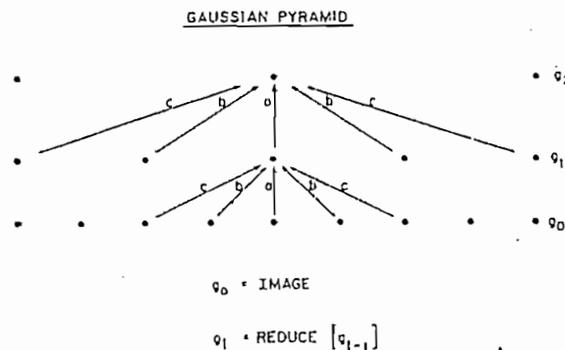
Este proceso será iterativo es decir, obtendremos un nuevo error de predicción L_1 al aplicarle a la nueva imagen original $g_1(ij)$ un filtro pasa-bajos, así obtendremos iterativamente funciones de error predichos para n pasos; $L_0, L_1, L_2, \dots, L_n$. Según la implementación cada uno es menor que su predecesor por un factor de escala de $1/2$. Los resultados serán apilados en forma de piramide. El valor de cada nodo en la piramide representa la diferencia entre dos funciones gaussianas iguales.

El primer paso para la codificación de la Piramide Laplaciana es filtrar la imagen original g_0 a través de pasa-bajos, para obtener la imagen g_1 ; y decimos que g_1 es una versión reducida de g_0 , en que la resolución y el muestreo es decrecida. De igual manera se obtiene un g_2 de g_1 . Hasta tener $g_0, g_1, g_2, \dots, g_n$ llamado Piramide Gaussiana.

Suponga que la imagen es representada por un arreglo g_0 la cual contiene C columnas y R filas de pixels. Cada pixel representa la intensidad de luz correspondiente a una imagen de puntos por un entero I entre 0 y $K-1$. Esta imagen viene del fondo o nivel cero de la Piramide Gaussiana, y se dice que el Nivel 1 de la piramide contiene imagen g_1 , la

cual es reducción o un filtrado pasa-bajos de la versión g_0 .

Para ello cada valor del nivel 1 llamados g_1 son calculados como una ponderación media de valores del nivel 0 con una ventana de 5×5 . Cada valor en el nivel 2, representa a g_2 , y obtenida de los valores del nivel 1 por aplicación de igual patrón de ponderación. Una representación gráfica de este proceso está en el gráfico siguiente.



El proceso de promediación de nivel a nivel es desarrollado por una función de reducción.

$$g_k = \text{REDUCED}(g_{k-1})$$

la cual promedia, para niveles $0 < l < N$ y nodos i, j

donde: $0 \leq j < R_l$; $0 \leq i < C_l$

$$g_l(i, j) = \sum_{m=-2}^2 \sum_{n=-2}^2 (w(m, n) g_{l-1}(2i+m, 2j+n))$$

Aquí N referencia el número de niveles de la piramide mientras C_l y R_l son las dimensiones del l -avo nivel.

Note que el mismo patrón de ponderación 5×5 es usado para generar cada arreglo piramidal de su predecesor. Este patrón de ponderación, llamado la Generación Kernel, es escogido sujeto a influencias certeras. Por simplicidad nosotros hacemos w separable.

$$w(m,n) = \hat{w}(m)\hat{w}(n)$$

Siendo normalizada la función \hat{w} de longitud 5

$$\sum_{m=-2}^2 \hat{w}(m) = 1 \quad \text{tq: } \hat{w}(i) = \hat{w}(-i) \quad \text{para } i=0,1,2.$$

Esta estipulación de que todos los nodos de un nivel dado debería contribuir a la ponderación total ($=1/4$) para los nodos en el siguiente nivel superior, llamada contribución igual.

Sea $\hat{w}(0)=a$, $\hat{w}(-1)=\hat{w}(1)=b$, $\hat{w}(-2)=\hat{w}(2)=c$ en este caso contribuciones iguales requieren que $a+2c=2b$. Estas 3 condiciones forzosas son satisfechas cuando

$$\hat{w}(0)=a \quad \hat{w}(-1)=\hat{w}(1)=1/4 \quad \hat{w}(-2)=\hat{w}(2)=1/4-a/2$$

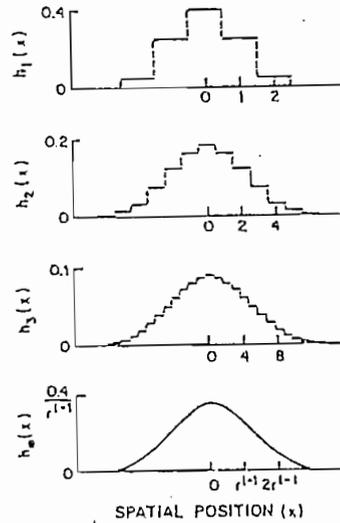
Generar una pirámide iterativa es equivalente a convolucionar la imagen g_0 con un conjunto de funciones de ponderación h_1 .

$$g_1 = h_1 \otimes g_0$$

$$o \quad g_1(i,j) = \sum_{m=-M_1}^{M_1} \sum_{n=-M_1}^{M_1} h_1(m,n) g_0(i(2^1)+m, j(2^1)+n)$$

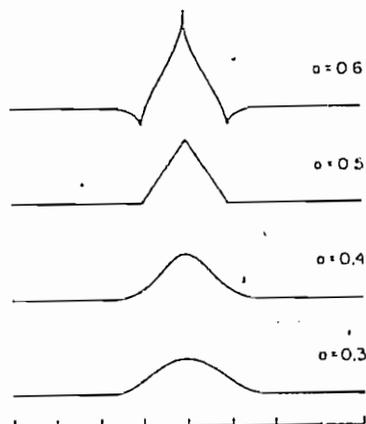
Si se dibujan los resultados de este procedimiento es decir, las funciones de ponderación equivalentes o niveles de pirámides gaussianas 1, 2 y 3, y con un $a=0.4$ tenemos el gráfico siguiente:

$$q_l = h_l \otimes q_0$$



Esta forma de funciones equivalentes convergen rápidamente a una forma característica con sucesivos niveles altos en la piramide, tal que solo su escala cambia. Esta forma depende del escogitamiento de α en la Generación de Kernel.

EQUIVALENT WEIGHTING FUNCTIONS



Las funciones de ponderación equivalentes son particularmente semejantes a las gaussianas cuando $a=0.4$. Cuando $a=0.5$ es triangular. Cuando $a=0.3$ este es ancho y llano. Con $a=0.6$ el modo central positivo es agudo y con picos y tiene lóbulos pequeños negativos.

Pero nuestro propósito es explicar las imágenes reducidas g_1 , es que estas pueden servir como una predicción de valores de pixels en la imagen original g_0 . Para obtener una compresión representativa, codificando la imagen de error la cual permanece cuando es expandida g_1 y sustraída de g_0 .

La piramide laplaciana es una secuencia de imágenes de error L_0, L_1, \dots, L_N . Cada una es la diferencia entre 2 niveles de la piramide gaussiana. Así para $0 \leq l < N$

$$L_l = g_l - \text{EXPAND}(g_{l+1})$$

$$= g_l - g_{l+1,1,1}$$

Donde g_{n+1} no es una imagen que pueda servir como imagen de predicción por g_n ; decimos entonces que: $L_N = g_N$

El valor de cada nodo en la piramide laplaciana es la diferencia entre la convolución de 2 funciones de ponderación equivalentes h_l, h_{l+1} con la imagen original.

Esto es similar a convolucionar un escalonamiento apropiado de funciones de ponderación laplaciano con la imagen. El valor del nodo podría haber sido obtenido directamente aplicando este operador, aunque esta consideración cueste una gran cantidad de computo.

Para la decodificación, la imagen se evalúa por expansión, sumando luego todos los niveles de la piramide laplaciana.

$$g_0 = \sum_{L=0}^N L_{L,1}$$

Un procedimiento más eficiente es expandir L_N y sumar esta a L_{N-1} . Luego de expandir esta imagen una vez, se suma esta a L_{N-2} , hasta el nivel 0, solo ahí es alcanzado y recobrado g_0 . Este procedimiento simple hace reversible los pasos en la generación de la piramide laplaciana.

$$g_N = L_N$$

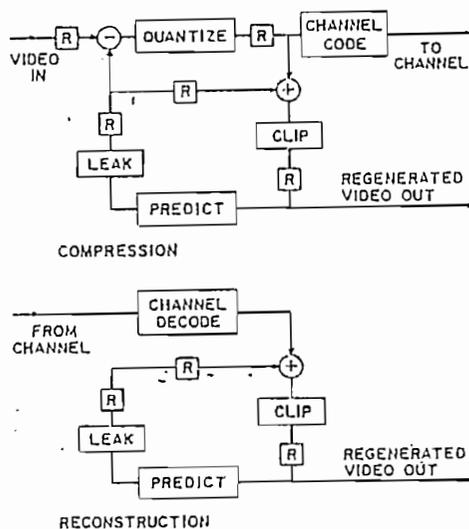
y para $L=N-1, N-2, \dots, 0$

$$g_L = L_L + \text{EXPAND}(g_{L+1})$$

Como es conocido las sumas son realizadas más rápido que las multiplicaciones (ya que estas son a base de sumas), es así que ganamos en tiempo de ejecución y menor cantidad de error en la operación del hardware.

2.3.- Administración de Memoria para un video codificado.-

Al abordar este tema no podemos dejar de topar la arquitectura como tal, de un circuito procesador de video, el procesamiento será diferente en su implementación y por ende la manera de guardar los datos en memoria. Así por ejemplo:



Este es el diagrama general de compresión y reconstrucción, usando codificación de predicción, sin importar el contenido interno de cada bloque.

La predicción es sustraída de la señal entrante y la predicción de error es cuantizada para los límites del conjunto de valores, en este caso hasta 16. La cuantización de error es devuelta para ser sumada con la predicción, y formar la señal reconstruida. Los valores reconstruidos anteriores son usados para generar cada predicción tal que en el receptor se puede usar el mismo circuito de predicción. El error cuantizado se manda bajo el canal para que en la recepción sea simplemente sumado con esta predicción y formar la señal digital reconstruida.

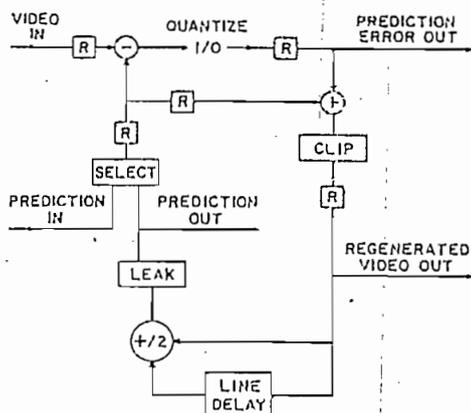
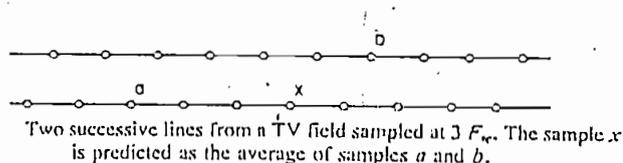
El circuito de CLIP en el codificador y decodificador previene sobre flujos o falta de este, que pueden ser

generados por la suma de predicción y de la predicción del error cuantizado. La función clip es controlada por la señal de predicción de error cuantizado. Para sobre flujo la salida es reemplazada por todos los ceros. Dando así, seguridad de inicio cuando recién se arranca la máquina, la predicción en el circuito es rápidamente asegurada sobre el nivel (cuadro) que existe en el circuito de compresión. Si un nivel equivocado existe en el circuito de reconstrucción se asegurará en un lazo lógico y permanecerá en el nivel hasta que la señal retorne. Esto normalmente ocurre con una línea o dos de T.V. El clip también ayuda a recobrar el bit de error cuando ocurre un overflow.

Generalmente, usando más muestras mejora la predicción, y por ello usamos una muestra de la línea previa, esto es definitivamente el mérito. La predicción seleccionada aquí es una media de el tercer muestreo previo sobre la línea que trasncurre y un muestreo de línea previa escogida para dar un balance de predicción. Esta predicción puede ser escrita así:

$$P(z) = (z^{-3} + z^{-5}b_1)/2$$

donde z^{-1} representa un muestreo retardado, z^{-3} representa tres muestreos retardados y así sucesivamente. Como ejemplo se muestra dos líneas sucesivas de un campo. La predicción de una muestra x es dada como la media de muestras regeneradas a y b .



Se puede usar un modelo con chips de 2.5 μ s y ajustados en 7 mm de área de silicón este circuito no incluye el canal cuantizador y codificador o decodificador; pues con el cuantizador y el canal de coder/decoder en external ROM, se mejora la compleja flexibilidad en escogitamiento de la cuantización y canal de códigos. El retardo es diseñado por un registro de desplazamiento, este consume más potencia lo que consumiría un circuito diseñado con RAM dinámicas. La implementación con RAM requiere una operación de lectura y escritura durante cada ciclo de reloj, haciendo más costosa y compleja su implementación.

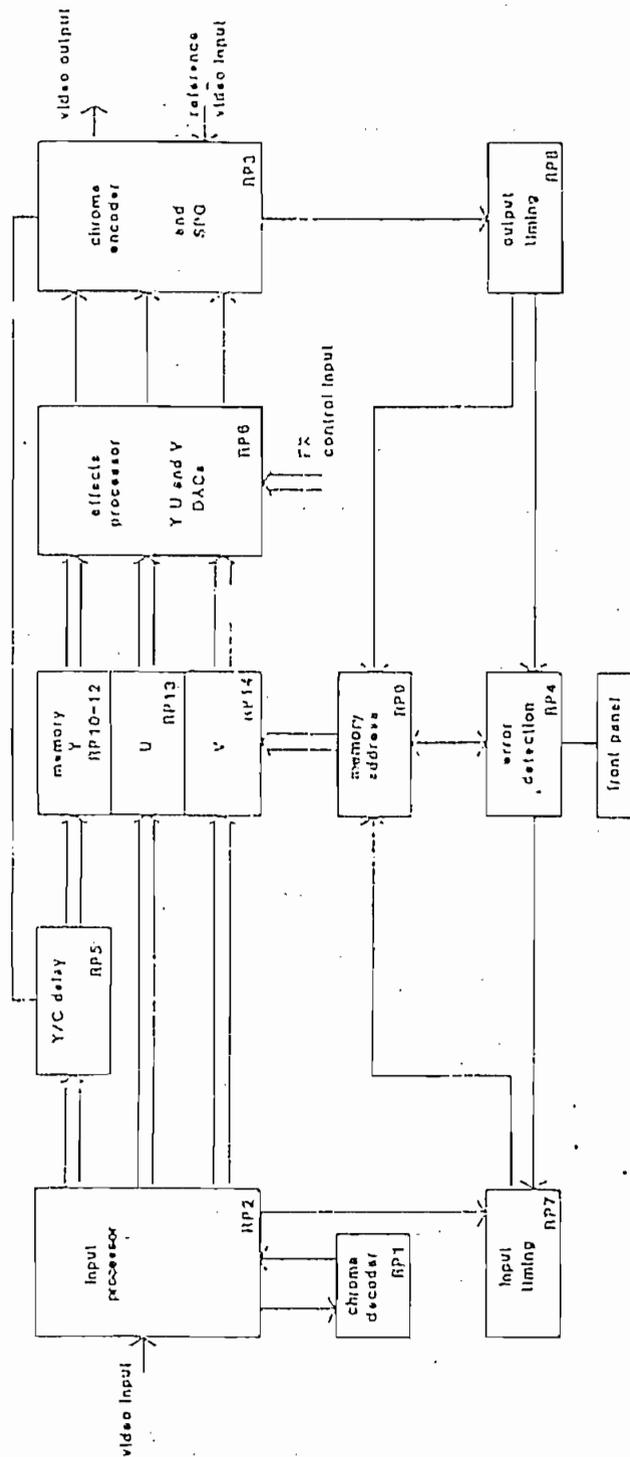
Aunque cada equipo tiene su particularidad en el proceso, todos tienen una similitud, guardando todos los distintivos y direcciones de la imagen en memorias y según

su sincronismo de lectura y escritura tendrán los efectos visibles, (esto en T.V.) como por ejemplo: la pixelación es para producir un mosaico por una técnica de sample-hold operando en el cuadro. La resolución de los efectos es continuamente variable de 112 muestras por línea, a 8 muestras por línea.

Para producir estos y otros efectos, la mayoría de arquitecturas deben contener:

- 1) Una entrada procesadora, la cual acepta video sin "sincronismo de enganche" (es decir, sincronismos que enfatizan la señal entrante o saliente con el oscilador interno de la máquina a una referencia externa que regula o sincroniza a todo el sistema o estudio, comunmente se los llama gen-lock), decodificando en sus componentes I, Q, y Y, y digitalizando cada uno de ellos; A estos se los llama sincronizadores, o un TBC (time base corrector).
- 2) Memoria de subsistema, comprendiendo un generador de direcciones y varias tarjetas de memorias que contienen memorias dinámicas usadas como entradas de video.
- 3) Por lo menos una salida del procesador, la cual después de recibir el contenido de la memoria del subsistema convierte esta a la forma análoga, y reconstruye en video compuesto.

- 4) Una entrada de regulación de tiempo de subsistema, la cual sincroniza la escritura del video de entrada digitalizado dirigiéndole hacia las memorias con sincronismo extraído de la señal original, tal que ordena la información recibida en líneas y campos para su procesamiento.
- 5) Una salida de regulación de tiempo de un subsistema, la cual sincroniza la lectura de datos guardados en memoria con SPG local (osciladores de burst-flag, línea y campo), la cual está normalmente en fase con gen-lock (llamado así a una oscilación o circuito oscilador que enfasa internamente a un equipo con una referencia externa) a una referencia de señal de video.
- 6) Un sistema de detección de error, el que detecta errores, principalmente en la entrada de video, e inicia una condición de congelado (se entiende por congelado, cuando un equipo mantiene en su salida un mismo cuadro o campo, sin renovar información en sus memorias, por falta de video inteligible en una de sus entradas del circuito procesador) por tanto tiempo como el error persista. Así para tener todas estas características el bloque general de memorias siempre estarán dependientes de como se concibió los demás circuito de proceso, por ejemplo:

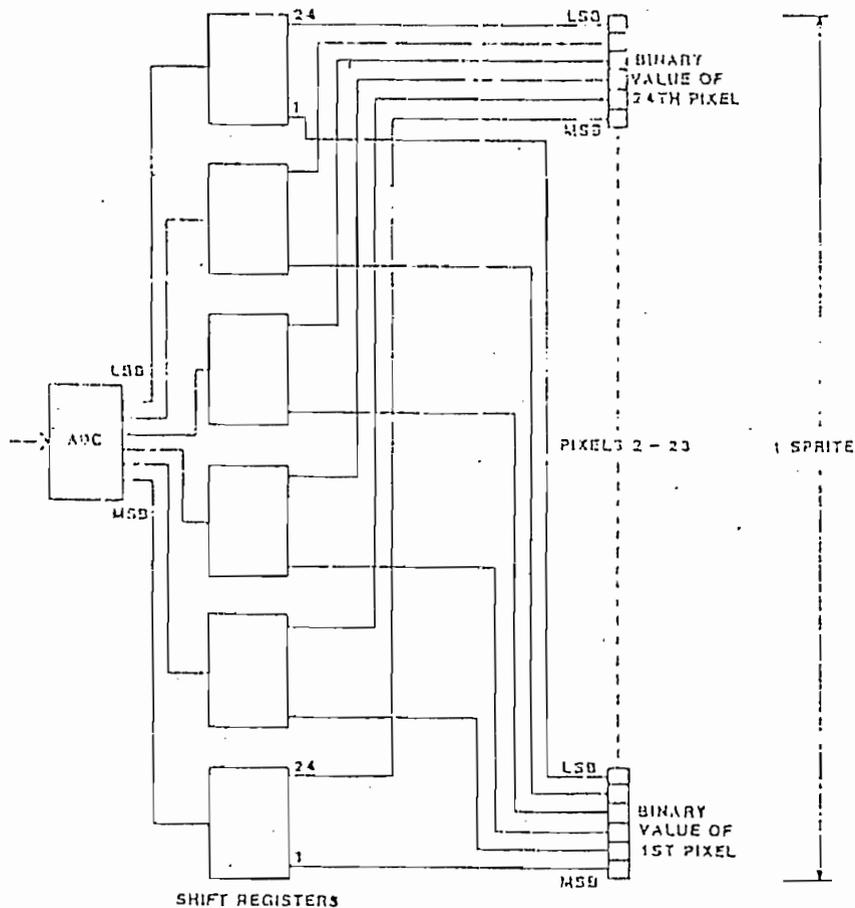


Para poder entenderse las condiciones anteriores se toma como ejemplo parte de las características de un equipo en especial, un sincronizador y generador de efectos tipo P147 CEL ELECTRONICS. Supongamos que la conversión A/D es de 6 bits y recordemos que filtrar Y es más rápido que filtrar el croma, esto por la respuesta de los filtros a bajas frecuencias es más rápido que a altas frecuencias.

La conversión A/D se hará simultáneamente tanto para Y, I y Q que guardarán estricta relación con los relojes de las memorias.

Si se clasifican a la palabra código por jerarquías de bits, existirán 6 bloques idénticos, uno por cada bit, y cada bloque tendrá las salidas necesarias, tal que contenga todas las combinaciones de bit que representan una línea, campo o cuadro. Así los registros de desplazamientos se llenan, cuando entran 24 datos LSB de 24 pixels, el grupo de 24 pixels se llama un "sprite".

.../...



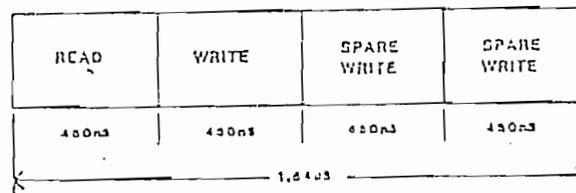
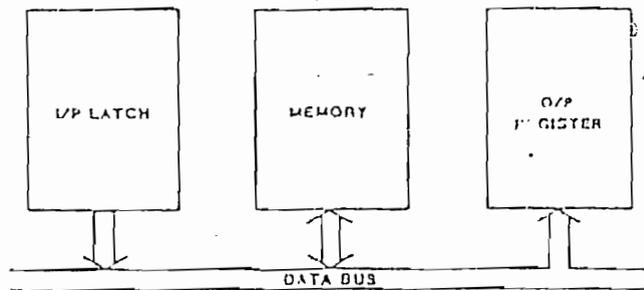
Esto nos implica 24 CK para llenar un registro. En cambio, 1 línea tiene 28 sprite de 24 pixels cada uno. La memoria para guardar un cuadro tendrá:

$$2(262.5)\text{lineas} \times 28 \text{ sprites} = 14700 \text{ direcciones}$$

Según los objetivos del equipo y/o algoritmos usados, podemos suprimir cierta información tal que ahorremos memorias costosas, sin degradar al video 'visible' como tal, al ojo humano.

Para crominancia (croma) se tiene que cada sprite solo contiene 8 pixels, esto implica que la memoria es un tercio que la de Y.

El manejo de las memorias se las hará con pulsos de handshake. Esto es necesario porque los buses de la memoria no son separados para lectura y escritura.



La manera de como vamos a ir ingresando los datos serán en forma paralela, es decir, todos los pixels primeros del grupo en los cuales se han dividido una línea o imagen, serán puestos en forma que las localidades de memoria sean equidistantes, tales, que al momento de ir sacando la información en efectos, estos sean rápidamente accesibles, como al fundir 2 videos, y dejar la "salida

serial" para efectos de wipe, es decir ir accedando una localidad después de otra, estas formas de accesos a memorias tanto en read/write es escogido según el efecto, grabadas en ROM las instrucciones que comandarán las señales de handshake.

Notemos que, tenemos alrededor de 2 Mbits; esto a pesar de no ser óptimo, e ir en contra del objetivo que lleva a la compresión del video, se justifica tan solo para equipos de procesamiento de señal interna, es decir, destinados a la no transmisión, sino a producir efectos para productoras de T.V. el problema se hace crítico cuando tenemos que transmitir esta señal a nivel comercial (broadcasting T.V.) o telemetría, en donde, los equipos para cumplir con las especificaciones y reglas internacionales serán demasiados costosas, y susceptibles a mayores errores.

El color, es una información que nos da mayor definición y conocimiento sobre la escena captada. El pero está que, como se dijo la abundancia de pulsos sobrepasa el ancho de banda permitido; entonces. La pregunta es como incorporar información sin aumentar tantos pulsos. Una de estas técnicas usadas, similares a las anteriores, trata con mayor detalle a la información de crominancia y a su mejoría respecto al ruido y su codificación.

2.3.1.- Compensación de color en codificación de imágenes.-

Con las técnicas de la NEC se reducen los requerimientos computacionales. Para compensar el color también debemos compensar cuadros en movimiento.

Una de las consideraciones de diseño en muchos bloques de aparejamiento en los algoritmos de compensación de movimiento es escoger el tamaño del bloque. Un pequeño bloque requiere una gran cantidad de gastos de información, porque se incrementa el número de vectores en movimiento que necesitan transmitirse, en vista de que grandes bloques resulta en un incremento de variaciones de errores de predicción, requiriendo así altas razones de datos para transmitir. Se han estudiado varias dimensiones de bloques 4x4, 8x8, 16x16, y fue encontrado que el bloque 8x8 provee el menor error medio-cuadrático de reconstrucción para una razón de datos dado. En la señal compuesta de color para T.V. la fase de color cambia 180° normalmente esto implica que el correspondiente pel (pel = punto de muestra de cada imagen) de dos cuadros sucesivos tienen 180° de fases de cambios; por ello puede usar tanto la señal compuesta o la componente de luminancia, la cual varía menos que el croma, pero si usamos señal compuesta tenemos altos grados de compresión y el segundo caso tiene menor ayuda computacional. La calidad de radiación de color NTSC y sus cuadros de prueba, suplidas por la NASA Lewis Research Center, Cleveland, se usan para evaluar el algoritmo.

El color compuesto es la señal de T.V. y esta es digitalizada a una razón de 4 veces la frecuencia de portadora de color, aproximadamente 14.32 MHz, con 8 bit información por pel. Excluyendo la parte de información de sincronismo, cada cuadro de imagen contiene 512 x 768 pel. Tres conjuntos de cuadros de prueba son suplidos con 3 cuadros sucesivos por cada conjunto. El cuadro de prueba es grabado directamente de los programas que son emitidos y contienen varias cantidades de objetos en movimiento, cambios de tonos, y flancos agudos.

Ensayemos un algoritmo para compensación de movimiento.

definamos:

B= Un bloque MxN de datos para ser procesado

R= correspondiente exploración de movimiento de B del cuadro previo.

la trayectoria del movimiento de cada paso es dado por el mínimo de la función de exploración de error $d(i,j)$ la cual es definida como:

$$d(i,j) = \sum_{m=1}^M \sum_{n=1}^N f[|b(m,n) - r(m-p_1(j), n-q_1(j))|] \quad (1)$$

donde $i = 1, 2, \dots, k$ número de pasos

$j = 1, 2, 3, \dots, 9$ la novena localidad

alcanzada por cada paso

b, r ---- denota los elementos de matrices B y R.

p, q ---- denota la localización de cada pel con respecto al centro del bloque donde es desarrollada la

exploración del movimiento.

f es una medición de función positiva y es definido para ser $f(.) = |.|$ como media estadística para un rápido computo. La dimensión de la matriz R es en pasos y puede ser expresada como:

$$[R]_1 = (M + VD_1 \times 2) \times (N + HD_1 \times 2) \quad (2)$$

donde:

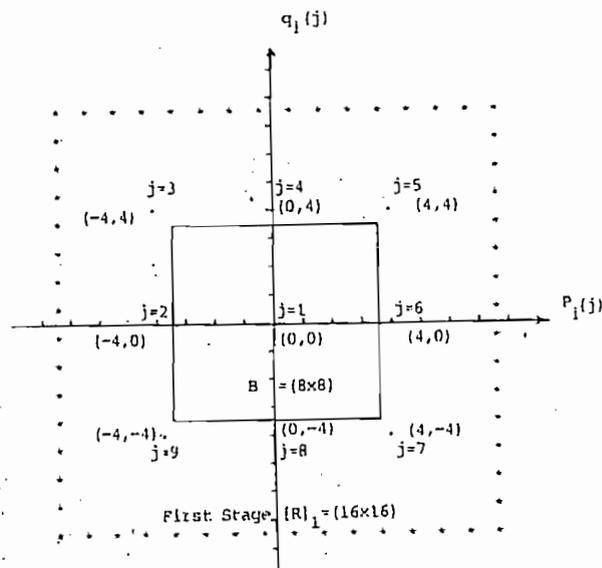
$HD_1 =$ desplazamiento Horizontal por un bloque
con respecto al centro de $[R]_1$

$VD_1 =$ desplazamiento vertical por un bloque
con respecto al centro de $[R]_1$

El rango de exploración de máximo movimiento (MMSR) tendrá un tamaño.

$$MMSR = (M + \max VD \times 2) \times (N + \max HD \times 2)$$

por ejemplo en la figura siguiente:



$i=4$ $B = (8 \times 8)$ $HD_1 = 4$ $VD_1 = 4$ $[R]_1 = (16 \times 16)$

$p_1(j) = 4, 0, \text{ o } -4$ y $q_1(j) = 4, 0, \text{ o } -4$

En pasos variables los algoritmos de búsqueda, $d(i,1)$ es primero computado, la cual corresponde a computar la suma del valor diferencia absoluta de pels sin ninguna compensación de movimiento. La suma calculada es comparada con $T1$, llamada umbral de tolerancia de ruido.

- i) Si $d(1,1) \leq T1$ esto indica que el bloque B es un bloque que no cambia. Este es un caso directo de intercuadro de DPCM.
- ii) Si $d(1,j) > T1$, la exploración de movimiento continúa a $d(1,2)$, $d(1,3)$, ... y $d(1,9)$. Luego se compara el computo $d(1,j)_{\min}$, $j=1, \dots, 9$ con el umbral $T1$.
- iii) Si $d(1,j) \leq T1 (j > 1)$, esto indica que es suficiente una sola ejecución de exploración de movimiento (stagnation search, SMS). Por eso es necesario una exploración posterior al movimiento, determinando así el correspondiente vector de movimiento para que localice el $d(1,j)_{\min}$.
- iv) Si $d(1,j)_{\min} > T1$ esto implica que un SMS no es suficiente. En este caso, la exploración de movimiento procede al segundo paso con una localización de $d(1,j)_{\min}$ como nuevo centro. Similarmente con el nuevo centro, el segundo paso de exploración de movimiento es computado usando (1) para determinar $d(i,j)_{\min}$ con $i=2$ y los valores apropiados $p(j)$ y $q(j)$.

- v) Si $d(2,j)_{\min} \leq T_1$, esto implica que dos SMS es suficiente y el vector de movimiento del bloque B se determina para localizar $d(2,j)_{\min}$.
- vi) Si $d(2,j)_{\min} > T_1$ y $[d(1,j)_{\min} - d(2,j)_{\min}] \leq T_2$, esto muestra que el bloque contiene excesivos datos ruidosos. Por cada bloque, un SMS probablemente resultará en un pequeño o nada de mejoramiento en terminos de exploración de error de movimiento. Computacionalmente, MS no precederá al siguiente paso en un bloque, y la localidad de $d(2,j)_{\min}$ es escogida para ser el correspondiente vector de movimiento.
- vii) Si $d(2,j)_{\min} > T_1$, y $[d(1,j)_{\min} - d(2,j)_{\min}] > T_2$, MS continúa al tercer paso. En el tercer paso de SMS, también son aplicados procedimientos computacionales y reglas de decisión similares al segundo paso. Si $d(3,j)_{\min} \leq T_3$ (también un mejoramiento del umbral), no propicia una exploración de movimiento y queda determinado el correspondiente MS.

Como quiera, si $k > 3$ y $[d(2,j)_{\min} - d(3,j)_{\min}] > T_3$, el MS procede al siguiente paso.

Usando este procedimiento, si el MS continúa hasta el último paso k , queda determinado el vector de movimiento correspondiente a $d(k,j)_{\min}$. Los avances de algoritmos en pasos variables de exploración de movimiento, se debe a que necesitan menos ayudas computacionales. Una de las ayudas computacionales es llamada submuestreo; con esta, solo un

submuestreo del bloque de pels se usa para computar los valores de diferencia absoluta de localidades de exploración de movimiento. En computación, usando $d(i,j)_{\min}$ y un submuestreo de 2 a 1, puede ser aplicado por el uso de otro elemento en el bloque en lugar de usar varios elementos. Para dimensiones de bloque de 8×8 , sin submuestreo todos los 64 elementos podrían ser usados para cada localidad de MS, sin embargo con el submuestreo de 2 a 1, se usa solamente 32 elementos. Obviamente con submuestreo de 2 a 1 y una adicional ayuda computacional de 50% en la computación de $d(i,j)_{\min}$.

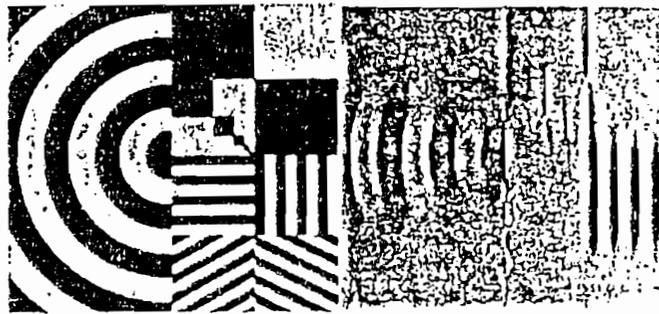
T_1 es el umbral de tolerancia de ruido; es probablemente el más importante umbral, pues ello nos da la calidad de imagen obtenida, estos valores que son estudiados y experimentados para cada aplicación, algunos ya están tabulados.

2.3.2.- Predicción de Imágenes.-

La predicción de imágenes, es parte misma de la modulación DPCM, también es particular del tipo de codificación. Así es la compresión por descomposición direccional; tenemos que la correlación entre flancos y las componentes vecinas sugieren que el diseño de un modelo predecible para estimar un flanco en una dirección de imagen dando sus vecindades espaciales en otra dirección de imagen.

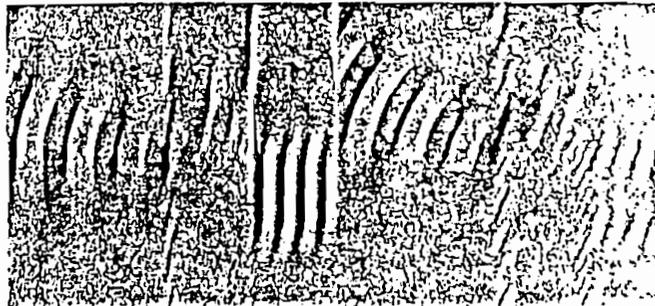
Analizando el siguiente ejemplo:

Dadas las primeras cuatro imágenes direccionadas después del detector de flancos. Los flancos correspondientes a la primera componente direccional (dirección número 1) está en (a). Estos flancos son detectados como flancos duros sobre un umbral alto y un flanco débil sobre un umbral bajo; estos son representados por negro y gris respectivamente. Los flancos en las vecindades de la dirección, también son sobre los mismos umbrales, los cuales son detectados como flancos duros en sus direcciones pero ellos vienen a ser débiles en las direcciones vecinas. En (b) flancos fuertes son detectados en componente número 2, mientras que a la misma vez son detectados las componentes número 1 y 3 como flancos débiles. La misma aplicación convencional para (c) y (d). Empezando de un flanco fuerte de la figura (a) la cual tiene en las vecindades unos flancos débiles, siguiendo la correspondiente curva en las figuras (b), (c) y (d) los flancos fuertes de la figura (c) pueden ser predichos de un flanco fuerte de la (a) y (b) usando un modelo predictivo de segundo orden. Luego el flanco fuerte (d) puede ser predicho de un flanco fuerte de (b) y el flanco predicho en (c). Como esta propiedad es limitada para curvas particulares (círculos, espirales), es necesaria una extensión consistente en una estructura general de predicción.



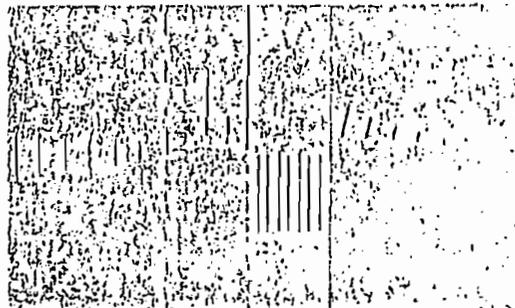
(a)

(b)



(c)

(d)



(a)

(b)

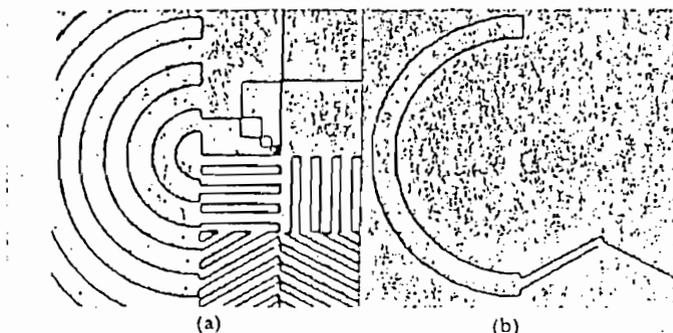


(c)

(d)

La estructura de predicción puede ser definida considerando los flancos de imagen (a). En esta imagen, es

posible seguir la curva formada por los segmentos adyacentes propios de las diferentes imágenes direccionadas. Tales curvas son llamadas estructuras de predicción, una de estas están en (b).



Una estructura de predicción es un conjunto de N flancos vecinos $E(n)$, $n=0,1,\dots,N-1$, cada flanco $E(n)$ es un flanco de imagen direccional particular y tiene una dirección denotada por $D(n)$. El modelo de predicción es definido sobre esta estructura. Si $X(n)$ es un vector k -dimensión caracterizando el flanco $E(n)$. La predicción 2-D consiste en definir el valor de predicción $x_p(n)$ de $x(n)$ como sigue:

$$x_p(n) = \sum_{i=1}^M a(i) \cdot x(n-i)$$

donde M es el orden de la predicción y $a(i)$ son matrices de $k \times k$. El error de predicción es

$$e(n) = x(n) - x_p(n).$$

Los parámetros de predicción $a(i)$, son obtenidos por resolución de las ecuaciones:

$$\sum_{i=1}^M a(i) \cdot c(i,j) = -c(0,j) \quad \text{para } j=1, \dots, M$$

donde:

$$c(i,j) = \sum_{n=n_0}^{n_1} x(n-i) \cdot x(n-j)$$

Hay MK^2 ecuaciones con MK^2 incógnitas, $x(n)$ debe contener pocos parámetros como sea posible para evitar muchos cálculos por un valor grande de k . Se ha escogido una descripción sintética de flancos $E(n)$ por $x(n)$, las primeras dos componentes de $x(n)$ son sus longitudes $L(n)$ (longitud del flanco) y sus direcciones $D(n)$. Con las coordenadas del punto de comienzo $S(n)$, permiten una exacta reconstrucción de la posición de los flancos $E(n)$. Pero las coordenadas de $S(n)$ son iguales para estos al punto final de $E(n-1)$, y por esta razón no es necesario incluirlos en $x(n)$. Las otras componentes son relacionadas para la descripción de los flancos de los contornos de los nivel de gris. Hay una magnitud y una componente de dispersión para el modelo de onda. Finalmente, son necesarios dos (o tres) parámetros si es usado un polinomio de primero (o segundo) grado.

C A P I T U L O I I I

3.- Implementación de un prototipo de un digitalizador de una señal de video.-

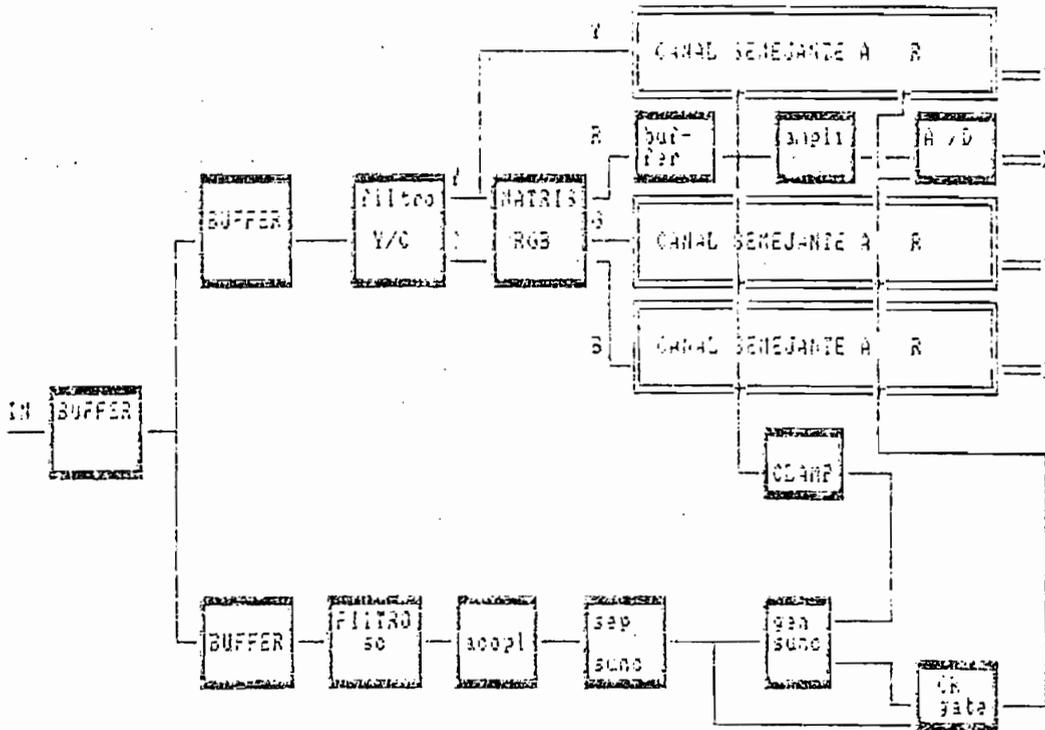
El escogitamiento de esta implementación como parte de un circuito más grande es tomado en base a que la conversión análogo-digital es muy semejante en la mayoría de los equipos es decir los pasos previos para que la señal ingrese al ADC, difieren muy poco uno de otros.

Además la implementación se la puede hacer para digitalizar el croma (RGB), y/o luminancia (Y), esto depende del tipo de tratamiento que se le vaya a dar a la señal que es muy propia y particular a modelos y marcas.

En el presente trabajo se realiza la digitalización de luminancia (Y), evitando implementar el digitalizador para croma ya que se lo obtiene por tres canales iguales a (Y). Aumentando así su versatilidad, pero también su costo. Este es un factor limitante para realizar, solo el canal (Y).

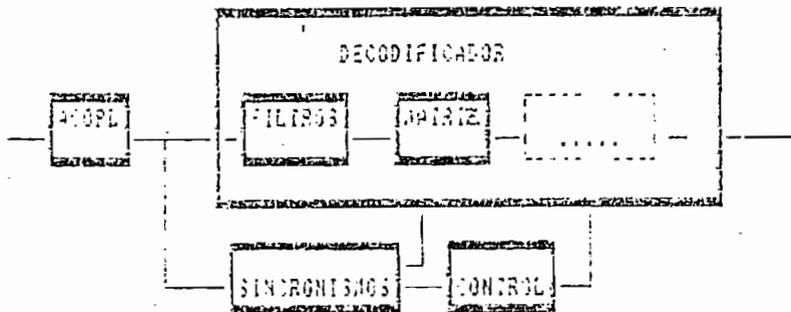
3.1.- Diagrama de bloques.-

El diagrama general para la digitalización a color (aunque la implementación trate de un solo canal (Y)), es la siguiente:



3.2.- Diseño y justificación del circuito.-

Enfoca el porqué del ordenamiento del circuito y sus razones. Se ha mencionado tres, por su generalidad en el tratamiento de la señal de video.



3.2.1.- Codificador.-

El codificar la señal implica cambiar su polarización, niveles de DC, separar sincronismos o invertirlos, digitalizar etc. Así podemos mejorar códigos digitalizados, cambiando de un código a otro para su mejor almacenaje y/o transmisión, es decir en vez de tener 1Ls en un blanco, tener 0Ls; mejorando potencias de disipación de los circuitos e interferencias. La separación y/o inversión de sincronismos, imposibilitaría la piratería de videos y manipuleo de la señal más fácilmente.

Podemos decir que codificar es alterar la señal de video en forma controlada y pre-establecida. Para este cometido se usa uno de los varios métodos que es el filtraje.

3.2.2.- Filtros.-

Estos son parte del gran bloque que se lo denomina codificador, ya que por medio de estos separamos

informaciones de la señal compuesta de video, así los sincronismos, que serán los que se usan para fazar los relojes de los circuitos, de memoria, selección de señales entre video1, video2....video n y la administración de señal e información en general.

Para este objetivo usan filtros pasivos o activos; los primeros muy conocidos que son sencillos y baratos pero que van en desmedro de la calidad del video por cuanto requieren un alto Q para su mejor cometido, logrando esto en límites estrechos. En cambio los filtros activos, realizan esto, procesando el video con operaciones de suma, resta, multiplicación retardos etc. de la señal, sin alterar la calidad del video original.

3.2.3.- Matriz.-

Este bloque, aunque no será implementado ya que implica separar el video en los 3 colores básicos, digitalizar por separado R,G,B y almacenar en bloques de memorias de 256 x 8, independientes los tres. Se muestra la manera de como se hará esta decodificación de la imagen en los 3 colores básicos (R-G-B).

Existen circuitos integrados en los cuales ingresan luminancia (Y) y crominancia (C) separados o como video compuesto, obteniendose a su salida las señales (R,G,B.) Recordemos que con Y y con las componentes de crominancia I,Q también transmitidas, podemos obtener los tres colores

básicos o las señales diferencia de color, según sea el caso, es decir; estas señales lo obtenemos decodificando la señal de crominancia; pues estas son modulación QAM de las señales I, Q, que a su vez son función de R-G-B, igual que Y, así:

$$I = 0.60R - 0.28G - 0.32B$$

$$Q = 0.21R - 0.52G + 0.31B$$

$$Y = 0.30R + 0.59G + 0.11B$$

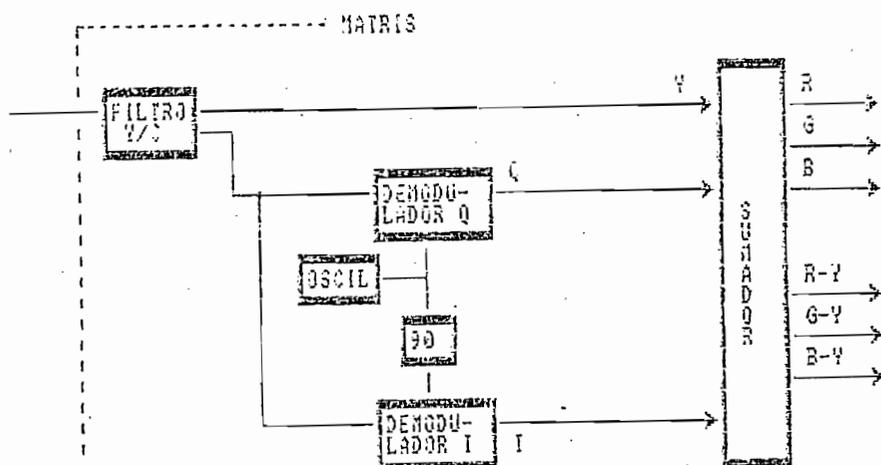
realizando operaciones algébricas se tiene:

$$(R-Y) = 0.948I + 0.62Q$$

$$(G-Y) = 0.271I - 0.63Q$$

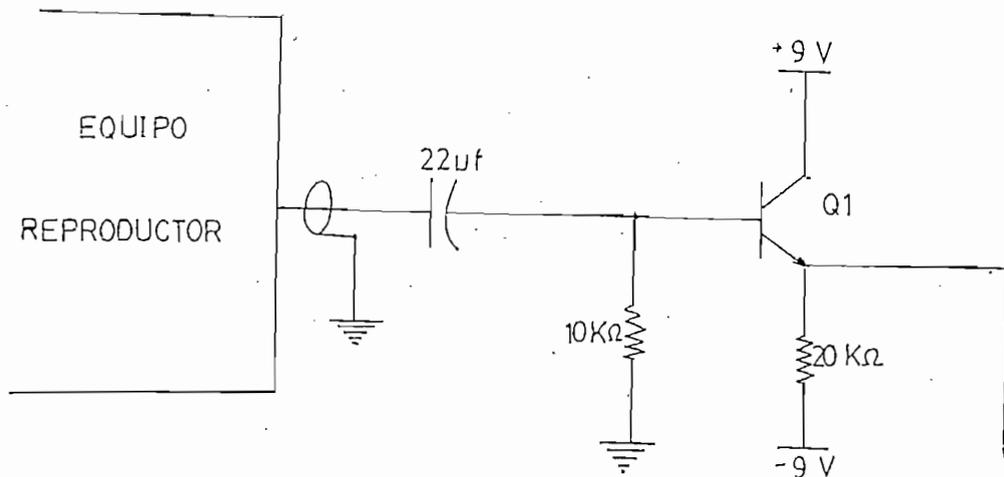
$$(B-Y) = -1.106I + 1.735Q$$

Entonces podemos notar que tenemos a disposición de nosotros los tres colores básicos o las señales diferencia de color con las tres señales que se captan en un receptor. Siendo permisible tener R' , G' , B' como señales puras.



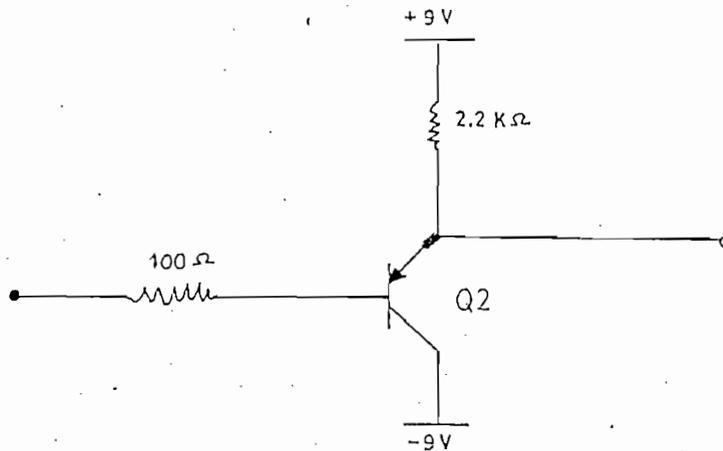
3.3.- Lógica de operación del circuito implementado.-

A la entrada del circuito se tiene un capacitor y al transistor Q1, los cuales acoplan polarización e impedancia de carga, esto es no cargue a la impedancia de salida del aparato que standarizado es 75Ω . La impedancia que observa la salida del equipo reproductor es $10 \text{ K}\Omega$ en paralelo con $\beta(2.2 \text{ K}\Omega)$, esto es practicamente $10 \text{ K}\Omega$, ya que la constante de ganancia de corriente del transistor usado es mínimo 100, que nos da $10 \text{ K}\Omega$ en paralelo con $220 \text{ K}\Omega$. Q1 sirve además de buffer de corriente pues alimenta a dos entradas, una para procesar el sincronismo y otra el video.



La información de video compuesto que tomamos de esta salida para separar de aquí los sincronismos necesarios pasa por un acoplador y buffer, el cual nos cambia de una baja impedancia a una alta impedancia sin aumentar el valor de la resistencia, aproximadamente a $220 \text{ K}\Omega$ con un $\beta=100$.

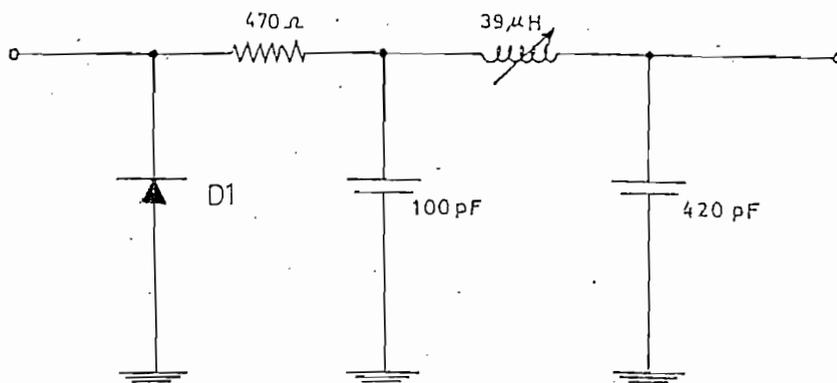
min.



La resistencia de 100 Ω sirve para evitar posibles oscilaciones. Notemos que no se ha puesto resistencias de gran valor para alcanzar la impedancia de entrada alta que sería lo más rápido de hacer, pero debemos tomar en cuenta que valores de resistencias altas introducen ruido y mal seguimiento de la señal en los flancos del video, esto es muy crítico, por cuanto la calidad del video se degrada, es así que utilizamos transistores como acopladores de impedancias y buffer. Es por esto que se encontrará en varias ocasiones más adelante.

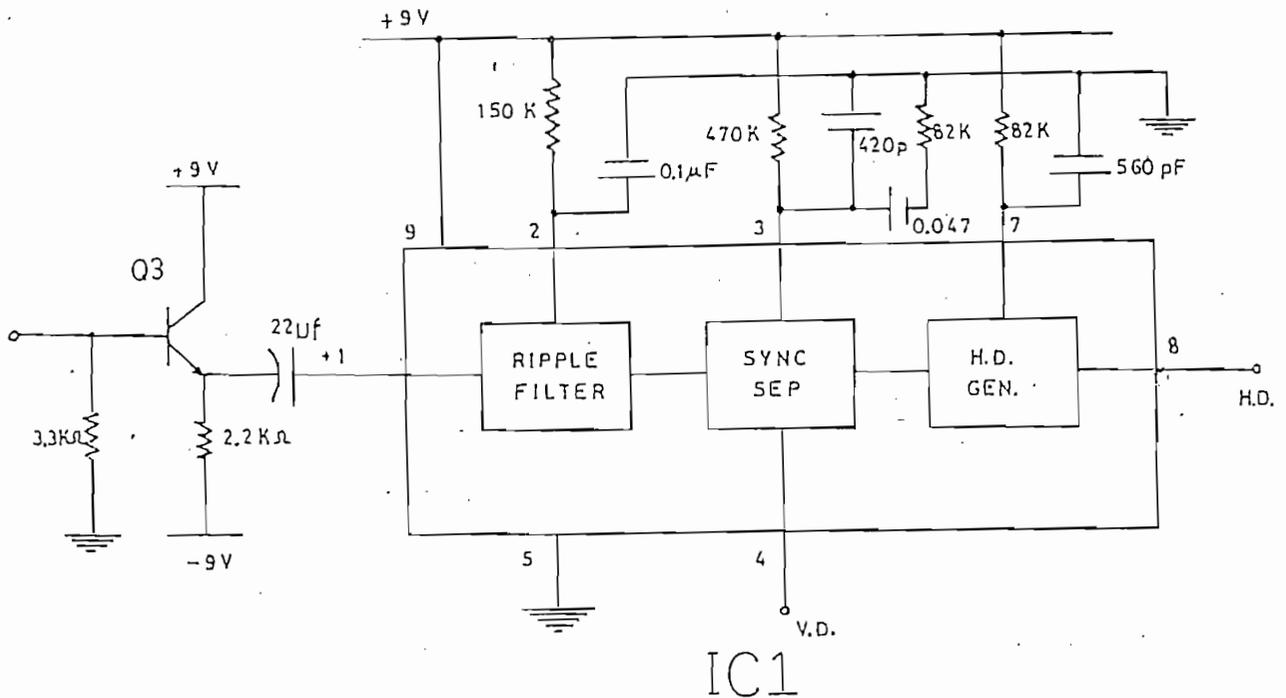
Nuestro siguiente paso es filtrar la componente de burst, que por ser una alta frecuencia respecto a la frecuencia de línea (3.58 MHz \gg 15.750 KHz) podrá causar ruido en los filtros más sensibles que están a

continuación. Este filtro lo hacemos con la inductancia $39 \mu\text{H}$ y los capacitores de 100 pF y 420 pF ; buscando el mejor diseño para filtraje y que no incremente la atenuación, nos llevó a usar los filtros de CAUER, en este caso usamos hasta un tercer orden. Suficiente para nuestro objetivo.



El diodo D1 es por protección de una saturación del transistor de la etapa anterior, pues si sucediera esto podría incrementar los voltajes en el filtro IC1. La resistencia de $3.3 \text{ K}\Omega$ se la usa para estabilizar más aún el buffer que se encuentra alimentando al IC1, como podemos ver a continuación:

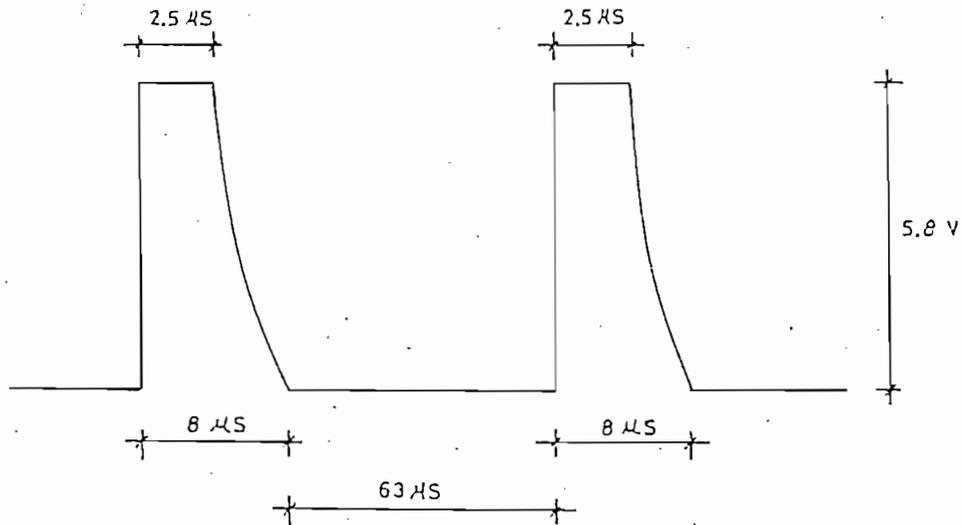
.../...



La resistencia de 2.2 K Ω es limitadora de corriente.

En diagrama de bloques de este filtro IC1 tenemos un 'Ripple Filter' (filtro de rizado), que como se dijo anteriormente el ruido en alta frecuencia 'relativa' desmejoraría el filtrado del sincronismo horizontal (blanking) y el sincronismo vertical, es por esto que se asegura que alguna espuria o en el peor de los casos el burst, no contamine al separador de sincronismo y a sus salidas de señal H.D (horizontal driver) y V.D (vertical driver). Las resistencias y capacitores son cercanos a los aconsejados por el fabricante para su polarización y mejor funcionamiento de este filtro que también es un generador

del pulso H.D. y V.D; las formas de ondas son las siguientes:



El sincronismo vertical también se obtiene pero incluido los pulsos HD.

El pulso HD obtenido del pin # 8 es alimentado al pin # 9 del IC2, el cual es básicamente un PLL (Phase Locked Loop), que sincroniza las oscilaciones de un cristal que alimenta a los pines 2 y 3, estos se alinean con el flanco anterior del blanking horizontal. Por cuanto el reloj que obtendremos del pin # 5 comandará al ADC (Análogo-Digital Converter), el cual tiene sus Sample-Hold incluido, que al digitalizar las líneas, todas tienen que empezar iguales, de lo contrario el patrón de conversión será, incorrecto; unas líneas empezaran con sample (muestreo o seguimiento de

la señal) y otras con hold (mantener información para la conversión), causando una sensación de desfase en la pantalla.

El cristal usado es de 14.318 MHz, valor que es un múltiplo del burst (3.579 MHz), de esta manera se lo tiene disponible en cualquier momento, solo dividiendo para 4 la señal de este cristal. Estas divisiones se logra con un divisor de frecuencia, flip-flop JK (IC3) obteniendo la frecuencia de reloj a 7.159 MHz si se divide para 2 y 3.579 MHz para 4. El primer reloj de 7.159 MHz sería usado para el ADC, tanto como el 3.579 MHz. El porqué no se usa el doble de frecuencia que el ancho de banda del video, es decir 12 MHz, y cumplir con las condiciones de Nyquist, está en que usamos filtros pasivos para filtrar la crominancia, ello disminuye en 2 MHz el ancho de banda de video para T.V., quedando en 2 MHz; según Nyquist tendríamos que tener una frecuencia de muestreo mínima de 4 MHz, pero por costos y asequibilidad a los C.I., se escogió el más cercano, tal que no se pierda información ni definición del video; así por fines demostrativos con una frecuencia de 3.58 MHz se tiene que:

$$3.583.5\text{MHz}/15.75\text{KHz} = 227.23 \text{ pixels por línea}$$

a $7.159 \text{ MHz}/15.750 \text{ KHz} = 454.54 \text{ pixels por línea}$. Esta cantidad de pixels tiene que cumplir con dos condiciones:

- 1) Tiene que tener una definición mejor o igual que la exploración por haz electrónico.

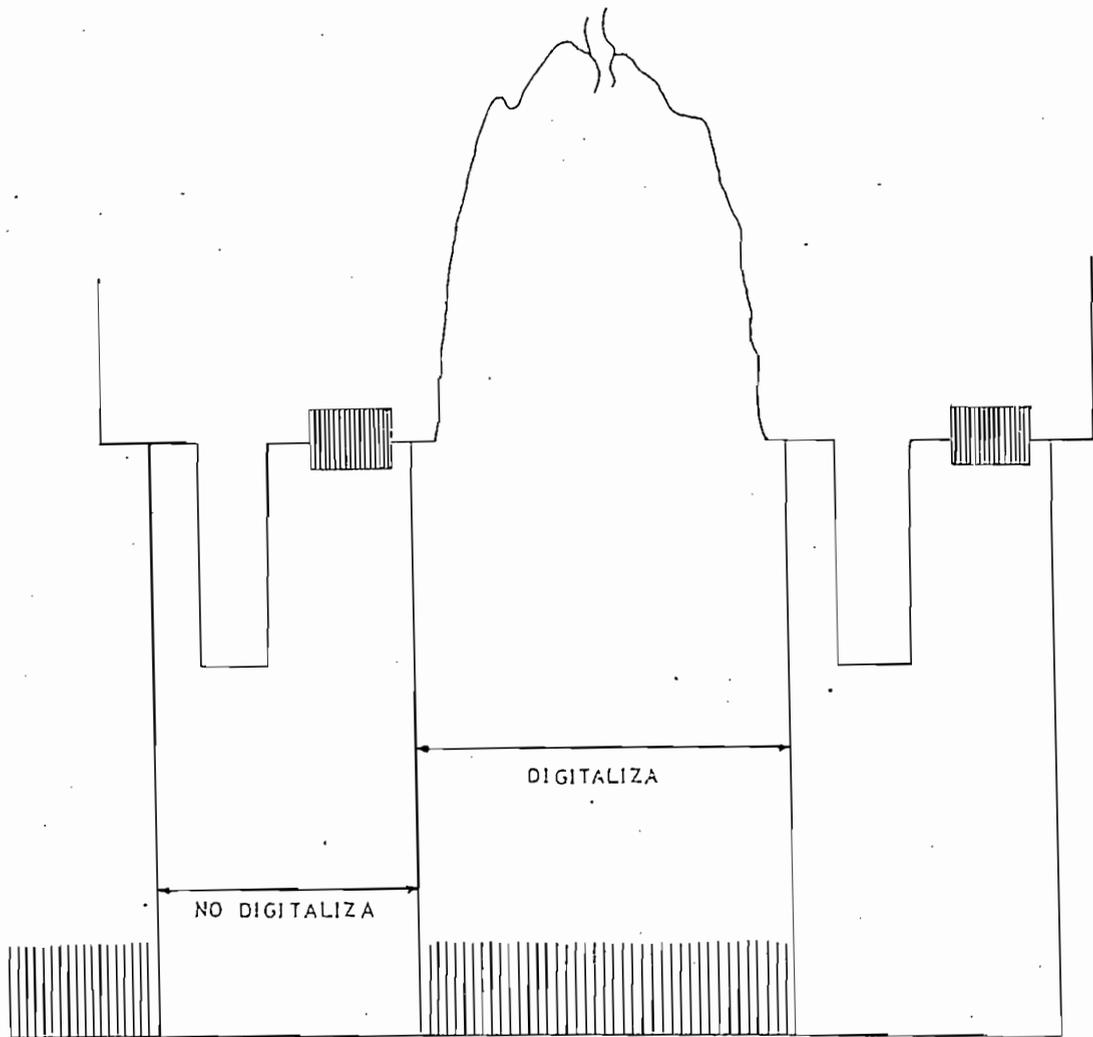
- 2) Sus sincronismos serán un múltiplo de frecuencias standarizadas, tal que su generación no se dificulte y no contamine con sus armónicos a todo el sistema; pues se tendría que llenar de filtros o choques al sistema.

Cada pixel es un cuadradito de la pantalla y obtiene un promedio tanto de luminancia o de crominancia, mientras más grande el cuadrado (pixels) la media es de una área mayor, perdiendo definición el video. Así los dispositivos de capatación de imagen CCD, disponen de pixels, que al ir sacando sincronisadamente forman una línea analógica; pero sí cada pixel en vez de ir sacandolo para unirlos en una línea de video se los va digitalizando, se tendría una "capatación digital". (recordando que el CCD es captación analógica, solo que la capta por celdas, elementos (cuadrados) de imagen analógico).

Para la frecuencia que trabajo de 7.159 MHZ el ADC funciona con sus especificaciones tope, para fines demostrativos y didácticos la definición desde 227 pixels/línea (< 358 MHz como CK) es aceptable, además de poder ser monitoreadas por equipos a los cuales se tiene acceso.

Lo que interesa es digitalizar la información de video, más no los sincronismos, de esta manera se ahorra potencia de disipación del ADC como contaminación del reloj en el sistema; para ello se usa "circuitos gate", los

cuales permiten insertar sincronismos o relojes, en espacios determinados en los cuales se desea el funcionamiento de ese, circuito, así tenemos burst-gate, clock-gate, etc..



Esto se logra con un circuito combinacional usando compuertas NAND:

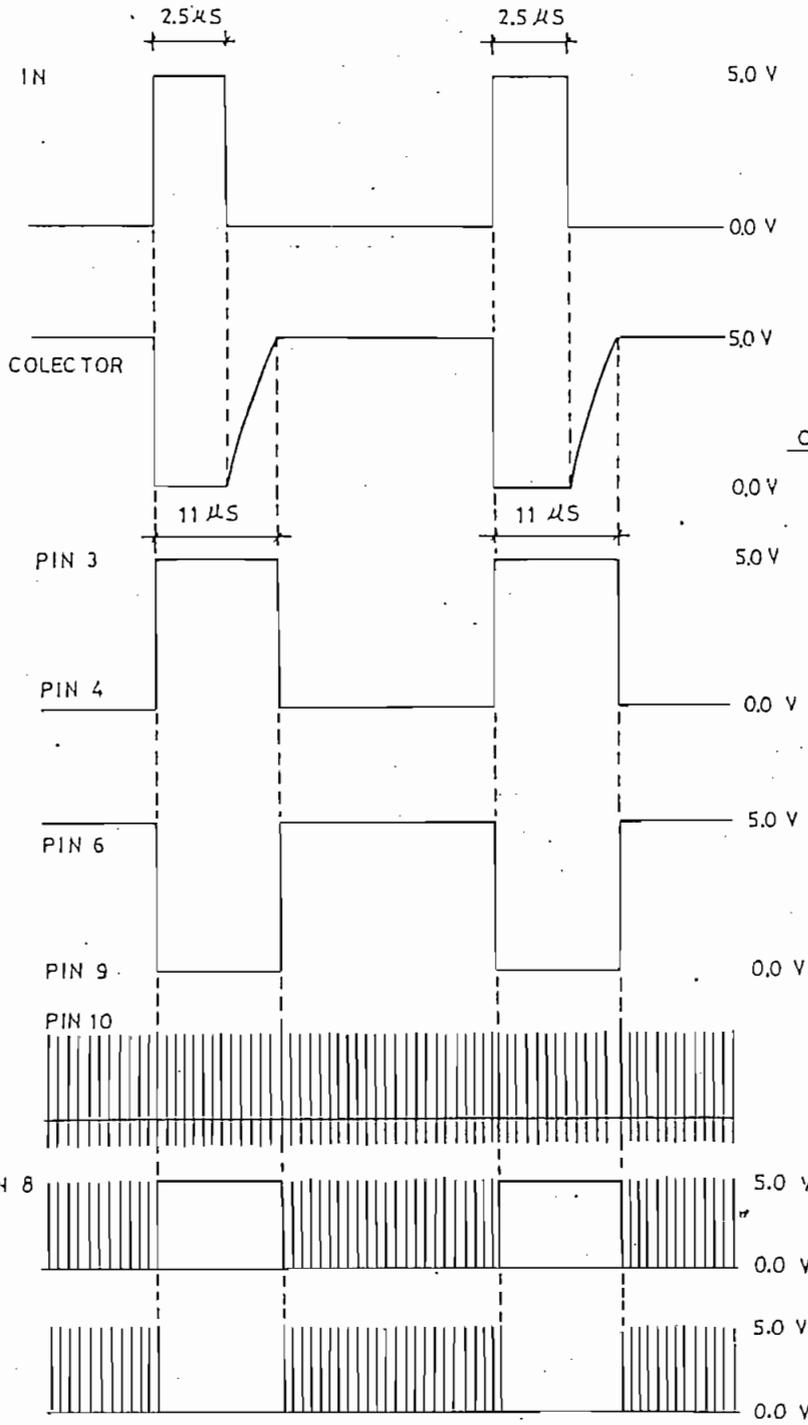
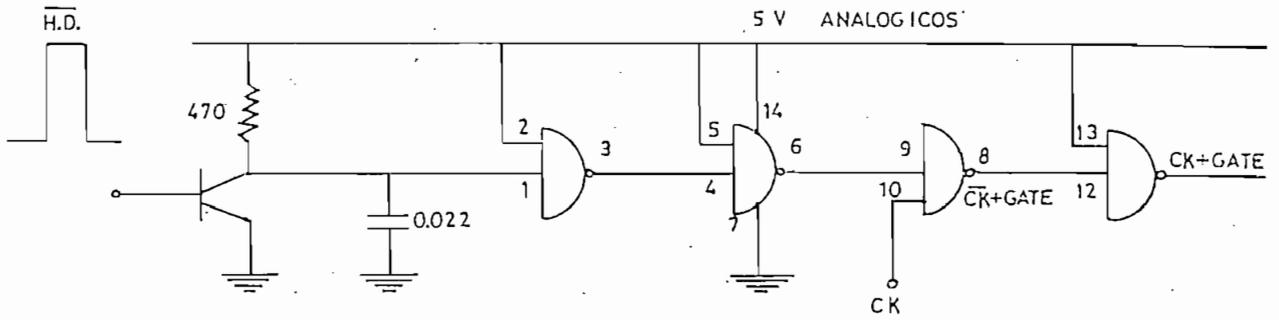


TABLA DE VERDAD
AND Y NAND

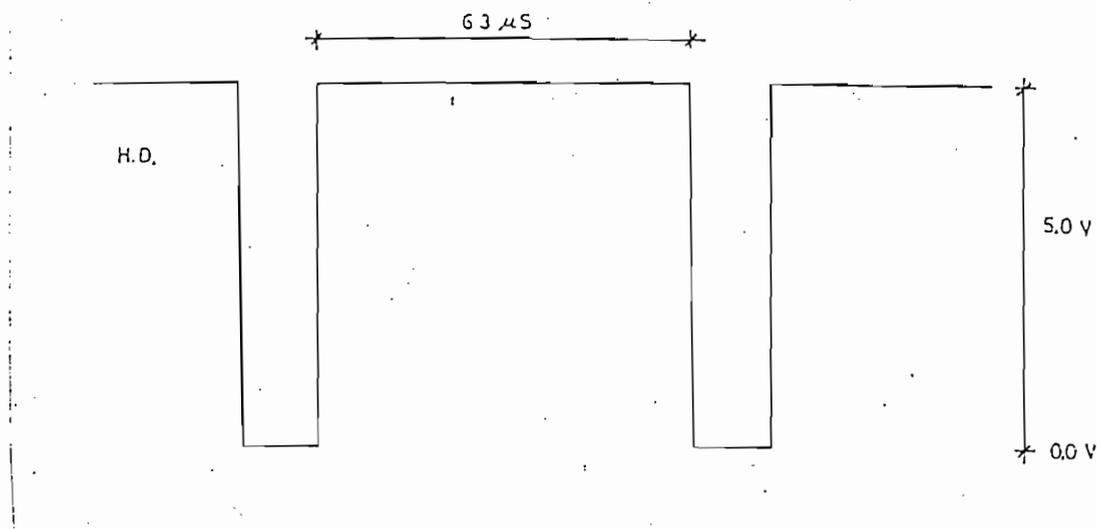
A	B	Q	\bar{Q}
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

CENTRAL	IN	Q	\bar{Q}
1	1	1	0
1	0	0	1

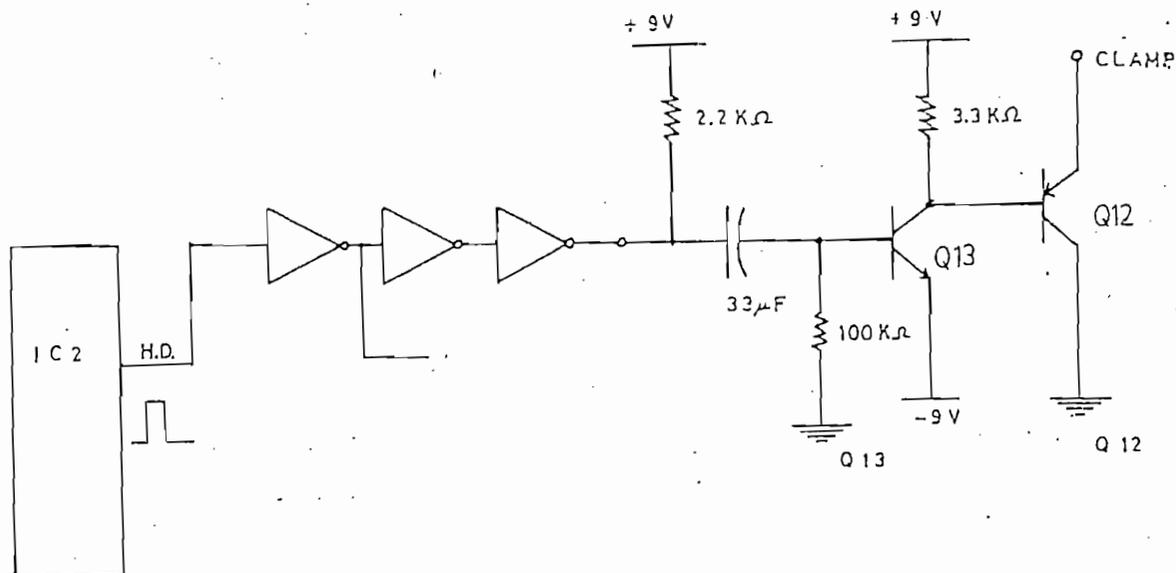
Se usa el capacitor para controlar la carga de este y por medio de ello el ancho del pulso, puesto que el período alto del pulso HD del IC2 en el pin 7 es de 2.5 μ s, necesitando los 11 μ s que tiene el sincronismo blanking o más, según sea el caso. Como la lógica de este pulso es positiva se ha invertido para alimentar el circuito gate; esta alimenta a una de las cuatro compuertas NAND, con la cual cuadramos el pulso, la segunda compuerta invierte la señal para tener la lógica correcta, es decir cuando la entrada está en 1 lógico permite el paso de la señal y en 0 lógico pone un 1 lógico en su salida sin dejar pasar la señal CK, (según la tabla de compuertas NAND), esta salida alimenta el pin 9 como señal de control de la tercera compuerta NAND.

La salida de esta es ya considerada como salida clock-gate, pero con lógica invertida; con la cuarta compuerta NAND actuando como inversor se tiene la lógica requerida para alimentar como reloj del convertidor AD.

El circuito clamping necesario para mantener la referencia de video en la digitalización. Es ejecutado en el período del blanking. Para ello usamos el pulso HD del IC2. El fan-out de la compuerta que nos da del pulso HD del IC1 no alcanza a manejar otras compuertas excepto la del IC2. Las características de este pulso son:



necesitando un pulso con una lógica invertida a esta, además para no cargarle a la salida del pin 7 usaremos 3 compuertas inversoras del IC6, del primer inversor tomamos señal para la base del transistor de entrada del circuito gate, y el fan-out de esta no abastece para otra base de transistor que además será usado como switch, es decir corte saturación; es por esto que tenemos que volver a invertir dos veces hasta tener la lógica correcta. En el tiempo de una línea de video tenemos cero voltios en el pin 6 del IC6 que se comunica por medio de la resistencia de $2.2 \text{ K}\Omega$ a $+9\text{v}$, existiendo una corriente hacia el pin 6 tal como lo indica el manual, en cambio en la base del transistor tendremos una corriente desde (tierra)- ($R=100\text{K}\Omega$)-(base-emisor)-(-9v), por ende el transistor Q13 estará en funcionamiento en la zona activa.

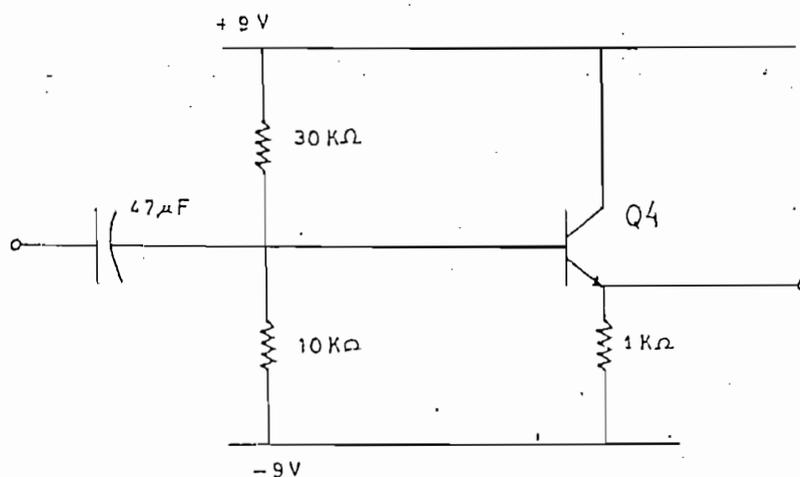


tal que en el colector del Q13 y base del Q12, tengamos un potencial de +5v de polarización, la parte del circuito que estará en contacto con el circuito clampig deberá tener una polarización menor a +5v, tal que esté en corte el diodo base-emisor, y por ende en corte el transistor Q12. En el período del blanking tendremos un pulso de 5.0v, este elevará el potencial de la base de Q13 a 5v lo cual le hará saturarse, la corriente que suplirá este estado será dado por tierra y -9v, pasando por la resistencia de 100 KΩ y juntura base emisor, ya que el capacitor tenderá a cargarse por medio de la resistencia de 100 KΩ, que también será el camino para la corriente de saturación; este a su vez saturará a Q12 puesto que arrastrará la base a un potencial de -9.0v saturando al colector; de esta manera

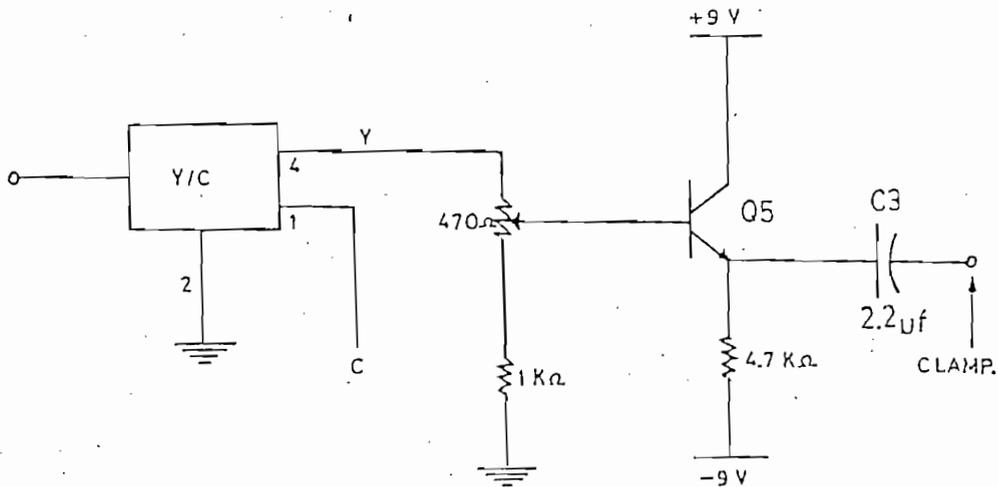
descargará el capacitor de paso del circuito del video que este conectado al circuito clampig. Así, de esta manera el circuito clamp "alisa" las señales cuando se tiene un DC parásito, restituyendo la referencia del circuito de video; en nuestro caso es 0.0v; pero no siempre pues se puede simular un fuente de voltaje variable tal que podamos referir a otro nivel de referencia.

Se ha visto hasta aquí el funcionamiento del circuito sincronizador, falta por explicar el circuito del video propiamente dicho, y este es así:

Se coge la misma señal de entrada, es decir video compuesto, haciendolo pasar por un circuito buffer de corriente y acoplador de impedancia, estabilizado por divisor de tensión de resistencias 30 K Ω y 10 K Ω .



Este buffer hará las veces de fuente de corriente para el filtro que a continuación se halla.

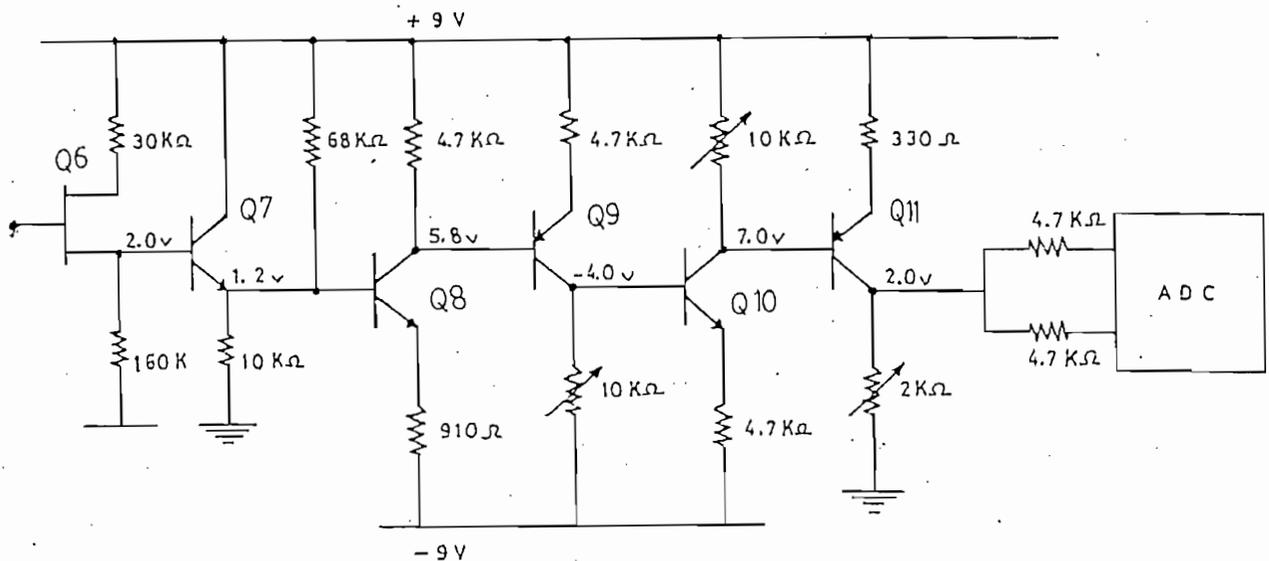


Como se trata de un filtro en circuito integrado, no indica su estructura interna, por experimentación se llegó a la conclusión de que el mejor funcionamiento se lo conseguía simulando una fuente de corriente.

Después de este filtro se tiene la señal de luminancia (Y) y croma separados. De estas se toma la señal (Y) y a través de un acoplador de impedancias y buffer de corriente para el capacitor de paso que se usará para clampear la señal. El cual debe ir antes del amplificador de video y de este al ADC. La señal de video se a atenuado de 1Vpp a 0.6Vpp aproximadamente. En este tramo del circuito se debe restituir el nivel de DC del capacitor de paso, consumiendo toda la corriente (positiva o negativa) de este, esto implica realizar un circuito que no permita corrientes de

otros circuitos aledaños. La solución está dada usando un transistor tipo FET canal n, donde se presenta alta impedancia de entrada (aprox giga Ω) y las corrientes son solo fugas; para dar camino a las equilibrantes ponemos una resistencia a tierra de 1.6 M Ω ; de esta manera toda la corriente usada será del capacitor C3, restituyendo su nivel de DC.

El circuito clamp se pone antes del amplificador, y en este ya no debe tener ningún capacitor de paso, sino habrá sido infructuoso el uso del circuito clamp.



El diseño de este amplificador no solo debe cumplir con una ganancia, sino que debe lograr un nivel de polarización en su salida que sea compatible o igual al offset del ADC.

Para nuestro caso el offset encontrado

experimentalmente es de 1.8v +/- 0.3v. este rango se lo puede calibrar colocando potenciómetros tal que varíe el nivel de polarización y poder además cumplir con la ganancia requerida.

El transistor Q7 sirve para mantener constante la corriente del FET y por ende su polarización, la resistencia de 10 KΩ sirve tanto para polarizar a Q7 como para la Q8, el cual nos da una amplificación de 5.0 aprox. Como la señal que teníamos era de 0.6 Vpp, en este punto tenemos ya 3.0 Vpp incluido los sincronismos. Nos interesa tener una amplitud que vaya desde 1.8 hasta 5.0 v solo de la porción de video, sin sincronismos, para que la conversión sea correcta y los blancos sean alcanzados al igual que los negros. Esto nos indica que como el sincronismo en el video compuesto tiene el 30% de la amplitud total (1.0 Vpp según indica la FCC) y el 70% será el video como tal; esto es 2.1 V, que al estar sobrepuesto al nivel de 1.8v llegaríamos a 3.9V. Con Q9 completamos esta amplificación hasta llegar a los 5.8 V, esto para tener un rango de calibración de los offset.

Pero, el nivel de polarización en su salida es de 4.0V. Tenemos que llegar con el video no invertido como lo está hasta ahora pero con nivel de DC de 1.8V mínimo, para ello usamos los transistores Q10 y Q11, que además de elevar este nivel podremos calibrar la ganancia y la polarización. Con este fin es lo que usamos potenciómetros. Cumplido esto, la señal es alimentada al ADC tanto para los

digitos MSB y LSB. La forma de como convierte de analógico a digital está en el desglose por bloques del ADC (anexo).

Los elementos usados en la implementación son:

- capacitores polarizados, y los de baja capacitancia son de cerámica.

- Resistencias son de 1/4 W.

- Los transistores son:

Q1 ----- ECG 108

Q2 ----- ECG 290A

Q14 ----- ECG 289A

Q13 ----- ECG 289A

Q12 ----- ECG 290A

Q4 ----- ECG 108/107

Q5 ----- ECG 108/107

Q6 ----- FET CANAL N; NTE 312

Q7 ----- ECG 108/107

Q8 ----- ECG 108/107

Q9 ----- ECG 106

Q10 ----- ECG 108/107

Q11 ----- ECG 106

CAPITULO IV

4.- Análisis de resultados y conclusiones:-

Según los resultados obtenidos en el proceso de elaboración del circuito en el presente trabajo, podemos indicar que:

- Armar un circuito en proto-board que trabajará a frecuencias en el orden de los MHz, este introduce capacidades parásitas. Influyendo directamente en el desarrollo normal del circuito, tal como, la degradación de la señal.

Los circuitos de sincronismo, que generan los relojes de mayor frecuencia, no oscilarán óptimamente por la existencia de las capacidades parásitas antes mencionadas, las cuales producen resonancia a otras frecuencias de los diseños, en unos casos, o inestabilidad por batimiento de la resonancia es decir, oscilan en un rango de frecuencias. El caso más común fué, que el circuito osciló con intermitencia. Esto obligó a armar por lo menos esta parte del circuito en un impreso prefabricado, donde el

funcionamiento fue inmediatamente corregido.

El problema de las capacidades parásitas han sido muy críticas, ya que casi todos los problemas se han centrado en ellas. Para poder disminuir lo más posible los efectos, se intentaron varias formas de armar el circuito, con la intención de minimizar las capacitancias, se prefería poner las polarizaciones tal que no coincidan en placas adyacentes, es decir los +9V a lado de -9V, sino que tengan menor diferencia de potencial entre los extremos.

Cabe notar que, los circuitos integrados usados para este objetivo no son comunes es decir, no confundir elementos integrados para un receptor de televisión ("la televisión") con elementos integrados para la televisión, los cuales son menos comunes aún que los anteriores.

A más de no estar disponibles en el mercado nacional tienen que ser obtenidos bajo pedido (a U.S.A. O Japón), por comparación de número de partes o repuestos en manuales o equipos.

Estos circuitos se los consiguió bajo pedido a desde SONY Miami; y son filtros Y\C, separador de sincronismo, generador de reloj y enfasador a pulsos de horizontal blanking y el conversor análogo digital. Además de ello la información técnica respecto a estos circuitos es tan limitada que explican el diagrama de bloques internos en el mejor de los casos, por lo general indican la distribución

de pines, y para circuitos con alta integración no es posible obtenerlas especificaciones, solo el número de repuestos (número de partes y modelo del equipo).

Este es el caso del generador de sincronismos del cual se tiene conocimiento de los pines por inspección y comparación de diagramas de varios equipos para procesamiento de señales de video. Sin embargo su funcionamiento distó del que se deseaba tener en el presente trabajo, teniendo que probar varias alternativas de conexión, algunos pines sin conectar y variando otros en sus elementos para oscilación y/o polarización hasta obtener el funcionamiento deseado.

Con el circuito integrado del conversor A/D (análogo-digital) se tiene la distribución de pines y el diagrama de bloques internos sin datos técnicos, y en el instante que se conectó al circuito no convertía a pulsos digitales, dudando de la veracidad de la distribución de pines, además de no saber el voltaje de offset, ni la corriente de alimentación que necesita el integrado. Todos estos parámetros fueron encontrados en forma experimental, así la corriente necesaria de alimentación es de 250 mA. La conexión de los pines de este C.I. con la cual funcionó correctamente es la que se indica en el diagrama del presente trabajo. El conversor fue escogido de entre otros en base a sus características extraídas por análisis de las especificaciones de equipos, tanto su frecuencia como su

capacidad de resolución. Y principalmente por el costo, siendo el más barato de los circuitos integrados usados en los equipos modernos, cuyos precios sobrepasan los 60 Dolares. El filtro Y/C es pasivo por ello solo tiene entrada, salidas y tierra. Para el caso del separador del sincronismo se tubo que ensayar con valores de elementos en su polarización hasta un buen desempeño de sus funciones.

Los transistores usados fueron obtenidos de tarjetas desechadas pertenecientes a equipos procesadores de video cuya numeración son:

- 2SC403C (tipo npn-Si AF)
- 2SA678 (tipo npn-Si AF)

Las características se logró tener del manual ECG. Para etapas amplificadoras o de paso de video ,se uso los transistores con numeración ECG 108 (nnp-Si RF/IF/Video Ampl, osc, mix, VHF/Uhf) y el ECG 106 (pnp-Si RF/IF Amp, osc, mix). Los elementos pasivos, como son resistencias, capacitores y potenciómetros se los ha adquirido en almacenes del ramo.

Se podría decir entonces que uno de los mayores limitantes en la elaboración práctica del circuito es el costo de los circuitos integrados, y la existencia limitada de ellos en el mercado.

Después de superados estos contra tiempos el circuito ensayado tiene un desempeño correcto, enmarcado en los limitantes que un proto-board introduce en su consecución.

Respecto a la fuente de polarización usada esta debe presentar buenas características de estabilidad, regulación y ruido. Por cuanto en varias sesiones de trabajo se usaron algunas fuentes de laboratorio, las cuales presentaron mala regulación y generan ruido, esto es a ± 6.0 V, causando problemas en el circuito en estudio debido al ruido de fuente; teniendo que ocuparse una fuente que no tenga ruido o tenga mejor regulación a ± 9 V., que es la polarización usada.

La conclusión global y total es que con la digitalización del video se ha incrementado la versatilidad y variedad de usos para procesamiento de la señal de video, influyendo esto en mayor variedad de efectos y baja complejidad de circuitos, pudiéndose incrementar la ayuda computacional en efectos y control del procesamiento del video.

Por presencia de las capacidades parásitas, inducción en circuitos, que influirán directamente en la calidad del video, además de la alta velocidad de ejecución que requiere el tratamiento del video, la industria especializada a ido integrando varios procesos dependientes o independientes del video en circuitos integrados monolíticos, siendo propio de cada empresa, es decir SONY fabrica sus Chips de diferente manera y disposición de pines de acuerdo a su conveniencia que difiere con los de otras marcas y de los estandares existentes en el mercado;

limitando por esta razón el desarrollo de aplicaciones prácticas de procesamiento de la señal de T.V. haciendo de esta manera que sea necesario contar con equipo especializado para tratamiento de video, en la que el desarrollo y aporte local, podrá ser de manera más puntual. La realización de este estudio teórico y desarrollo e implementación de un circuito para la conversión A/D de una señal de video, ha permitido vislumbrar, por una parte el enorme campo de aplicación y las perspectivas del procesamiento digital de señales de video, el cual este trabajo es solo el primer paso, y por otra las dificultades ocasionadas por limitaciones de orden tecnológico sobre el tema en nuestro país. Debo mencionar adicionalmente que no obstante que la industria de la T.V. comercial explota este servicio utilizando la mejor tecnología que esta a su alcance, considero que el acceso a esta tecnología desde el punto de vista formal y académico no nos es posible al momento, por lo que concidero se debe dar un mayor impulso y apoyo al desarrollo de ensayos relacionados con el tema.

Listado de un programa destinado a la reconstrucción de imágenes que han sido segmentadas.

IFS Decoding in BASIC

Listing A is a BASIC implementation of the random iteration algorithm. It includes the data for the Sierpiński triangle, but you can use it to process any IFS tables. In particular, you will want to try the data in tables 2, 3, and 4. Be sure to set the variable *m* correctly; it tells the program how many transformations are in the IFS.

It is also essential that the probabilities in *p*() add up to 1. For speed, the transformations should be listed in descending order of probability: the highest probability transformation first, and the lowest probability last.

The program includes variables for rescaling and translating the origin to accommodate the range of the points being plotted to the limits of your screen. If the image is too wide, decrease *xscale*; if the points are too close horizontally, increase *xscale*. Adjust *y*scale similarly to get a good vertical point spread. To move the image, adjust *xoffset* and *yoffset*.

You can do these adjustments by trial and error: Run the program; interrupt it and change the offsets and scale factors; and run it again. Or, you can replace the plot command *pset* with a command to print the values of *x* and *y* and run the program to get an exact idea of the range of points being plotted, so you can adjust the scale and offsets more precisely.

Another way to arrange the program is to have it read all the data—*m*, *a*(), *b*(), *c*(), *d*(), *e*(), *f*(), *p*(), *xscale*, *y*scale, *xoffset*, and *yoffset*—from a disk file specified by the user. Instead of reading in the coefficients *a*, *b*, *c*, and *d*, you may want to read in angles θ and ϕ and scale factors *r* and *s*, and then calculate the coefficients.

The random iteration method is computation-intensive, so we recommend use of a compiler such as Microsoft's QuickBASIC or Borland's Turbo BASIC. If your computer has a floating-point coprocessor and your compiler supports one, so much the better.

Listing A: A BASIC program demonstrating the use of the random iteration algorithm to reconstruct an IFS-compressed image.

```

10 'Allow for a maximum of 4 transformations in the IFS
20 DIM a(4), b(4), c(4), d(4), e(4), f(4), p(4)
30 '
40 'Transformation data, Sierpinski triangle
50 'First comes the number of transformations
60 'then the coefficients a through f and probability pk
70 'The values for pk should be in descending order.
80 DATA 3
90 DATA .5,0,0,.5,0,0,.34
100 DATA .5,0,0,.5,1,0,.33
110 DATA .5,0,0,.5,.5,.5,.33
120 '
130 'Read in the data
140 READ m
150 pt = 0 'Cumulative probability
160 FOR j = 1 TO m
170   READ a(j), b(j), c(j), d(j), e(j), f(j), pk
180   pt = pt + pk
190   p(j) = pt
200 NEXT j
210 '
220 'Set up for Graphics
230 SCREEN 3 'Select graphics screen
240 xscale = 350 'Map [0,1] onto [0,350]
250 yscale = 325 'Map [0,1] onto [0,325]
260 xoffset = 0
270 yoffset = 0 'Leave the y-origin
280 '
290 'Initialize x and y
300 x = 0
310 y = 0
320 '
330 'Do 2500 iterations
340 FOR n = 1 TO 2500
350   pk = RND
360   'The next line works for m<=4. It must be modified
370   'for values of m > 4.
380   IF pk <= p(1) THEN k = 1 ELSE IF pk <= p(2) THEN k = 2
           ELSE IF pk <= p(3) THEN k = 3 ELSE k = 4
390   newx = a(k) * x + b(k) * y + e(k)
400   newy = c(k) * x + d(k) * y + f(k)
410   x = newx
420   y = newy
430   'Use PRINT x,y instead of the PSET line
440   'to see the range of coordinates. Then fix
450   'xscale, yscale, xoffset, and yoffset
460   IF n > 10 THEN PSET (x * xscale + xoffset, y * yscale
           + yoffset)
470 NEXT n
480 '
490 LOCATE 24, 35
500 PRINT "Press any key to end.:"
510 WHILE INKEY$ = ""
520 WEND
530 '
540 'Return to text screen
550 SCREEN 0
560 END

```

ANEXO 2.-

En base a la descomposición direccional dado por el patrón de direcciones se logra obtener entre otros las fotos de imagenes optenidas por experimentación, que ha continuación se indican:

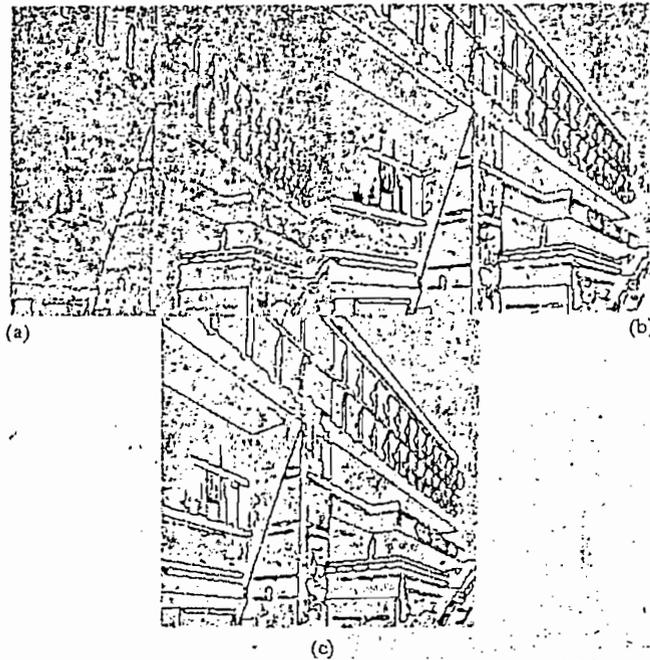


Fig. 17. Edge detection by magnitude thresholding. (a) Magnitudes of zero-crossings to be thresholded. (b) Result of a noncontextual thresholding. (c) Result of ridge finding.

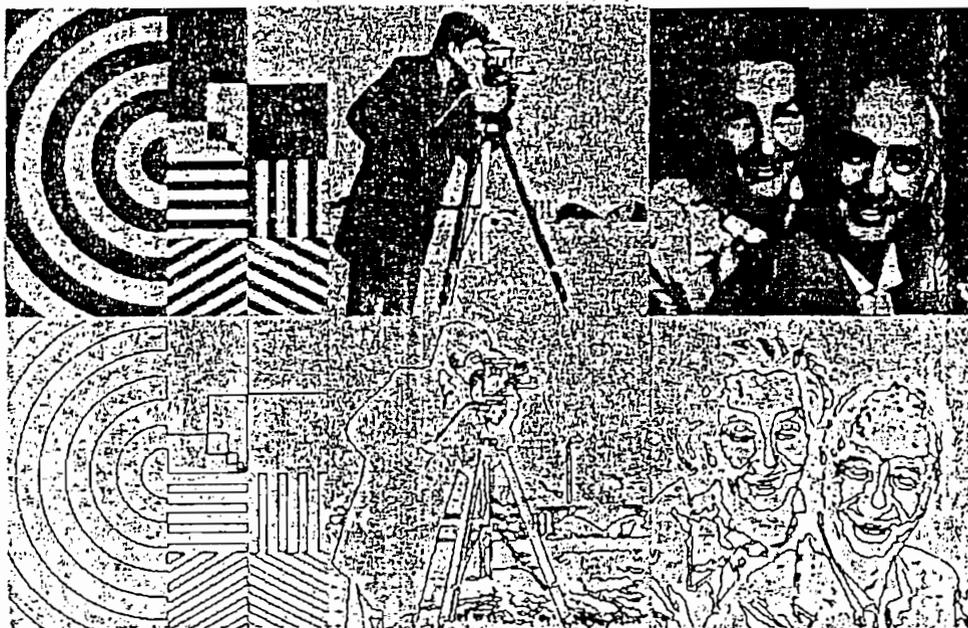
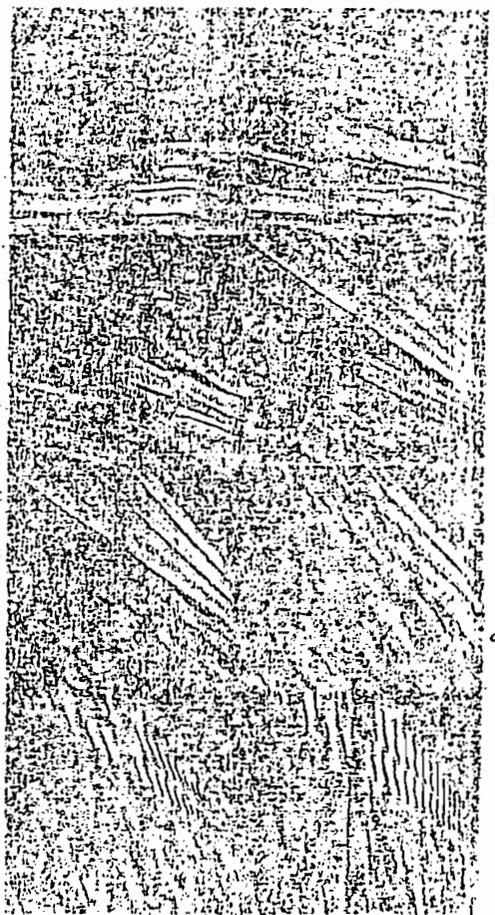
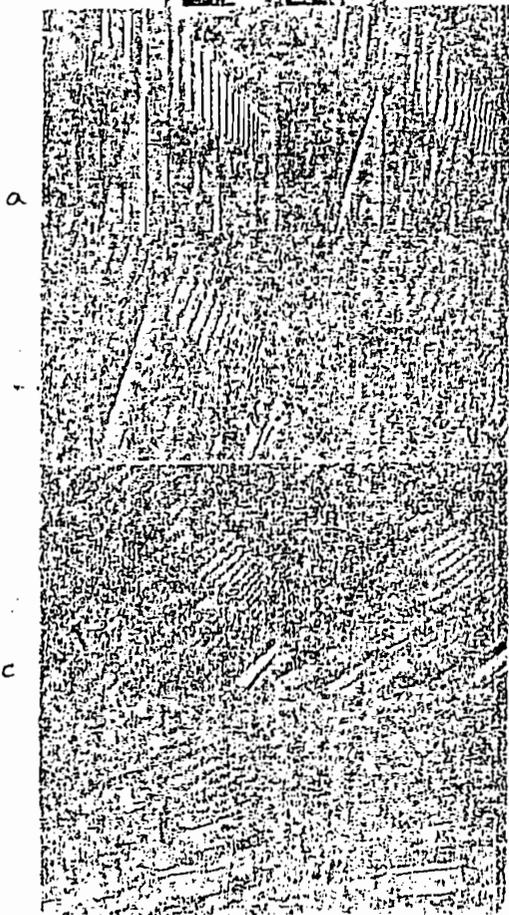
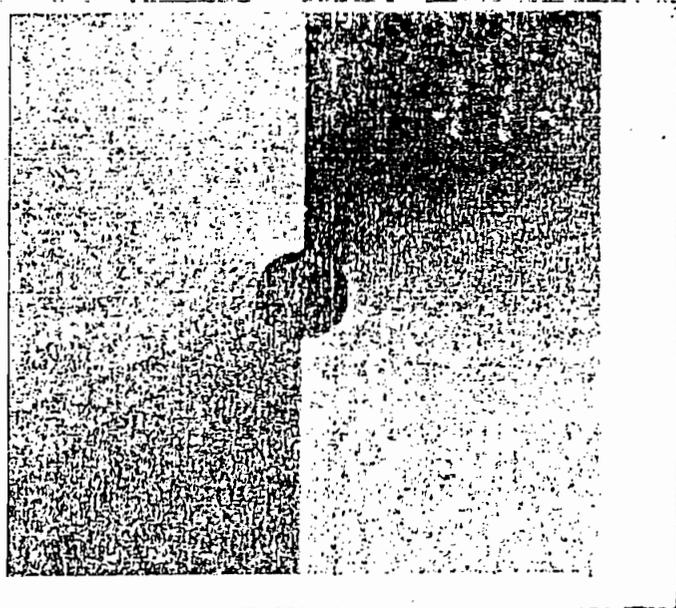


Fig. 19. Results of edge detection on three original images.



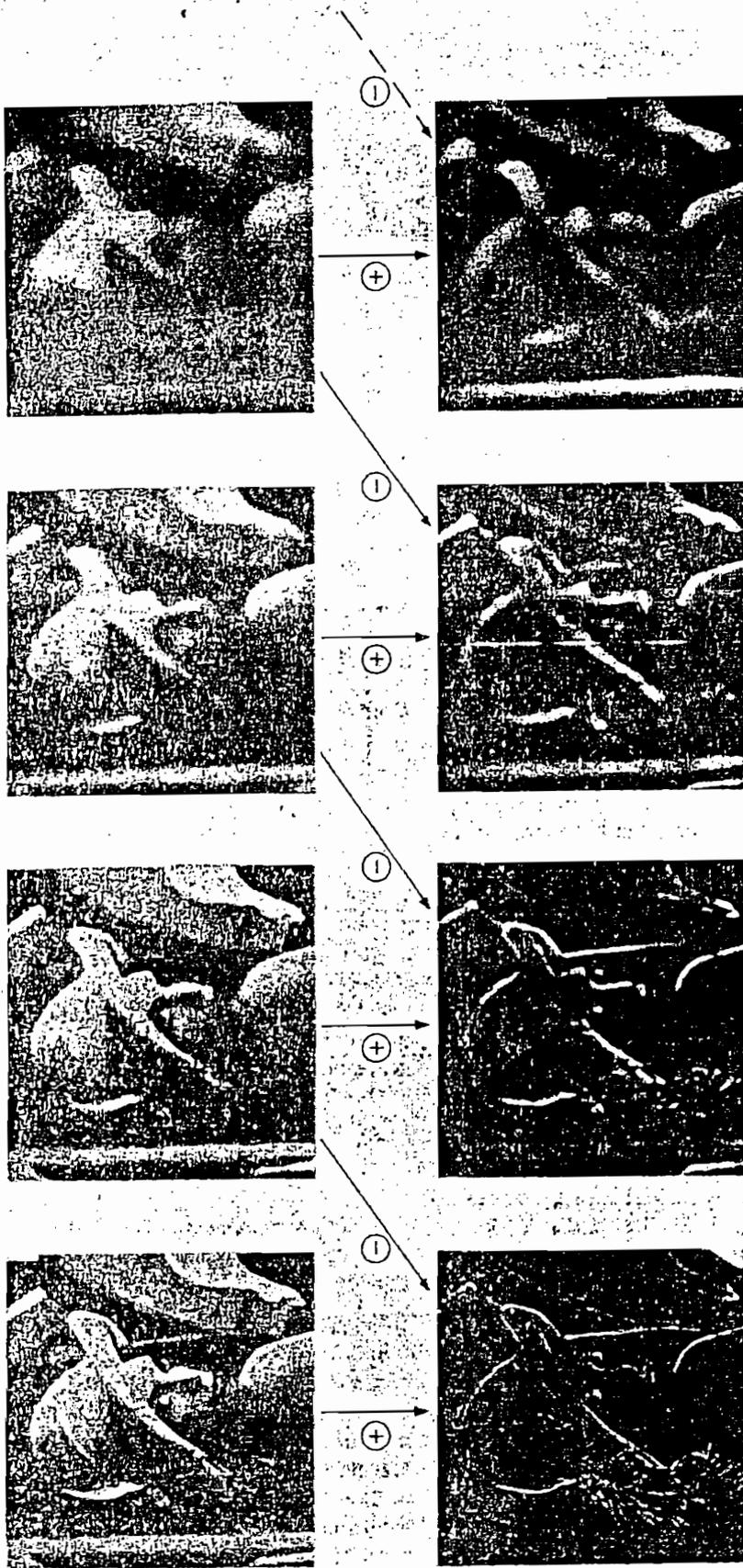


Fig. 5. First four levels of the Gaussian and Laplacian pyramids. Gaussian images, upper row, were obtained by expanding pyramid arrays (Fig. 4) through Gaussian interpolation. Each level of the Laplacian pyramid is the difference between the corresponding and next higher levels of the Gaussian pyramid.

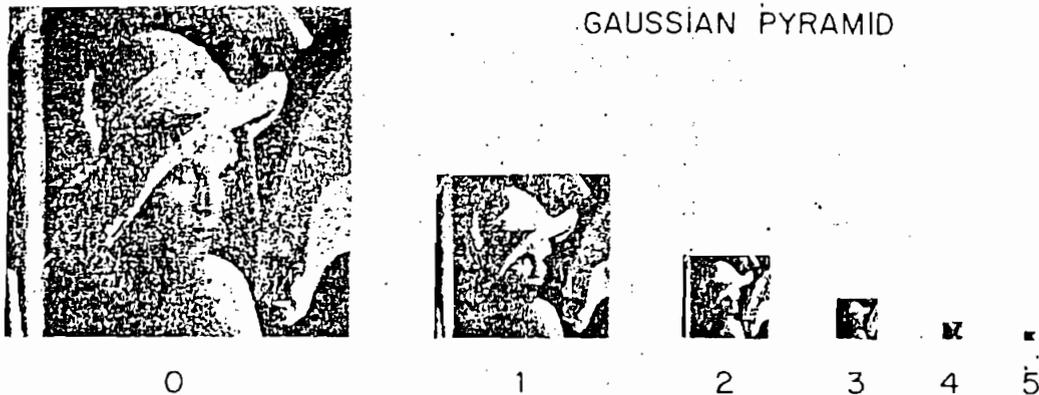


Fig. 4. First six levels of the Gaussian pyramid for the "Lady" image. The original image, level 0, measures 257 by 257 pixels, and each higher level array is roughly half the dimensions of its predecessor. Thus, level 5 measures just 9 by 9 pixels.

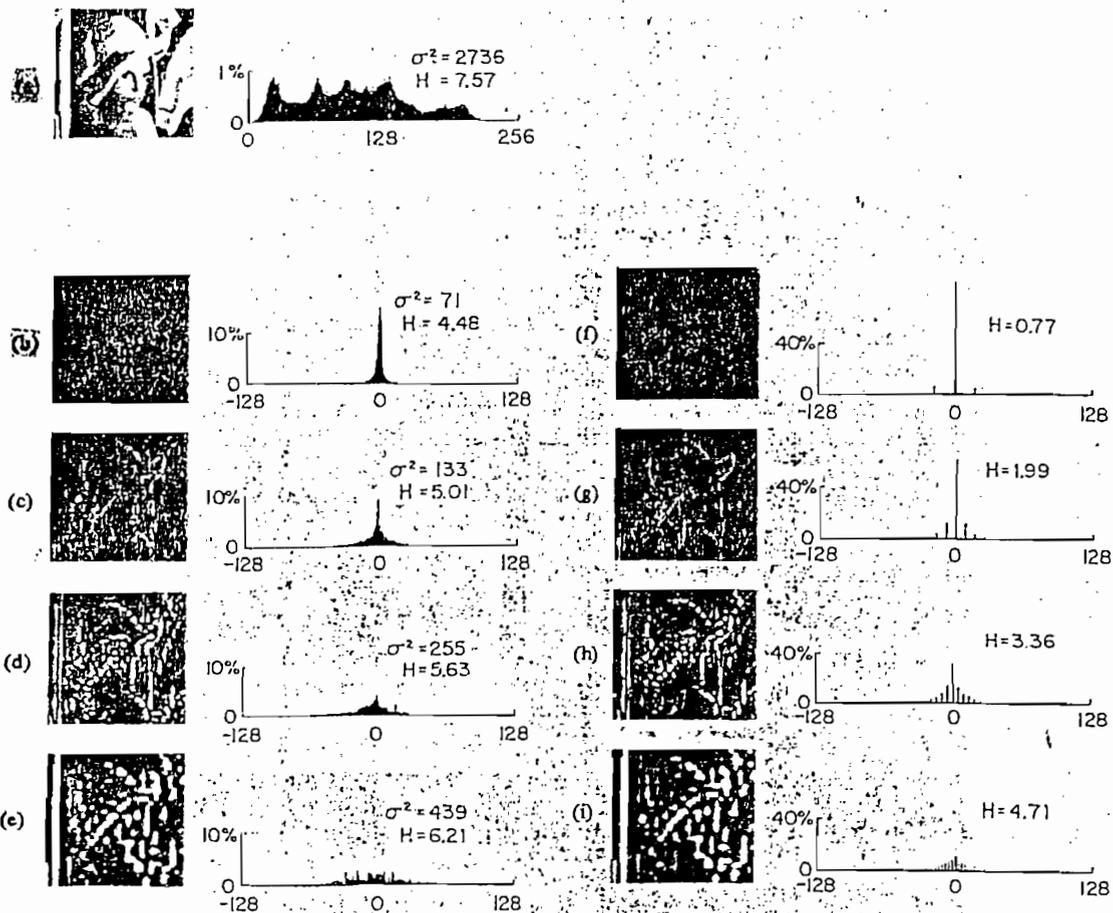


Fig. 6. The distribution of pixel gray level values at various stages of the encoding process. The histogram of the original image is given in (a). (b)-(e) give histograms for levels 0-3 of the Laplacian pyramid with generating parameter $\sigma = 0.6$. Histograms following quantization at each level are shown in (f)-(i). Note that pixel values in the Laplacian pyramid are concentrated near zero, permitting data compression through shortened and variable length code words. Substantial further reduction is realized through quantization (particularly at low pyramid levels) and reduced sample density (particularly at high pyramid levels).

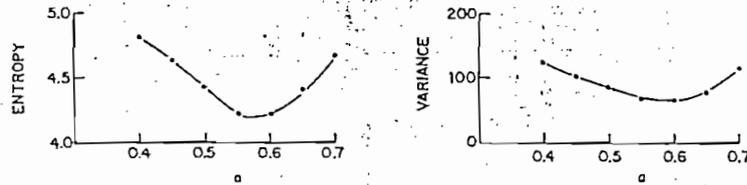


Fig. 7. Entropy and variance of pixel values in Laplacian pyramid level 0 as a function of the parameter "a" for the "Lady" image. Greatest reduction is obtained for $a \approx 0.6$. This estimate of the optimal "a" was also obtained at other pyramid levels and for other images.



Fig. 8. Examples of image data compression using the Laplacian pyramid code. (a) and (c) give the original "Lady" and "Walter" images, while (b) and (d) give their encoded versions. The data rates are 1.58 and 0.73 bits/pixel for "Lady" and "Walter," respectively. The corresponding mean square errors were 0.88 percent and 0.43 percent, respectively.

ANEXO 4.-

Predicción lineal.

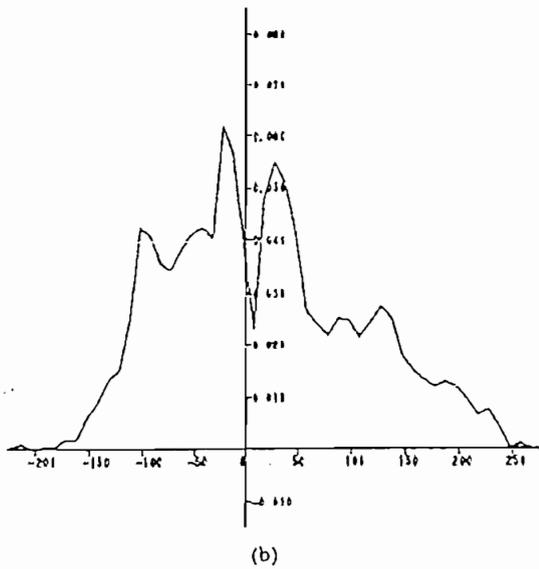
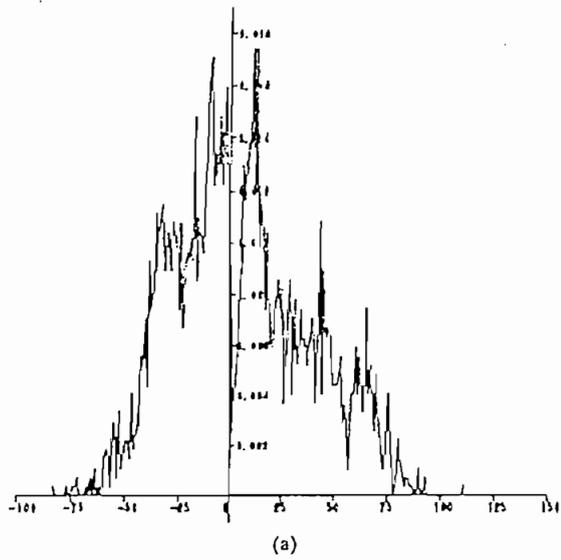


Fig. 32. Histograms of the approximation coefficients. (a) Synthetic high magnitudes and (b) wavelet magnitudes.

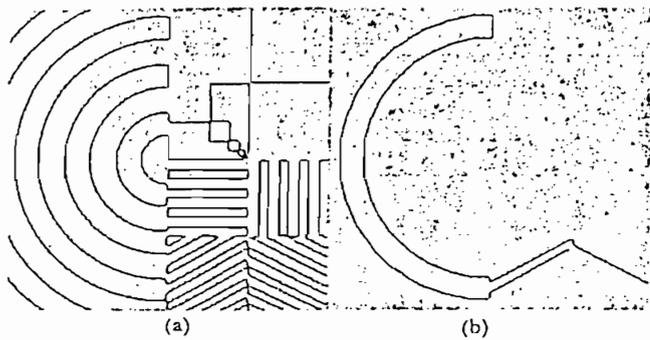


Fig. 35. (a) An edge image and (b) an extracted prediction structure.

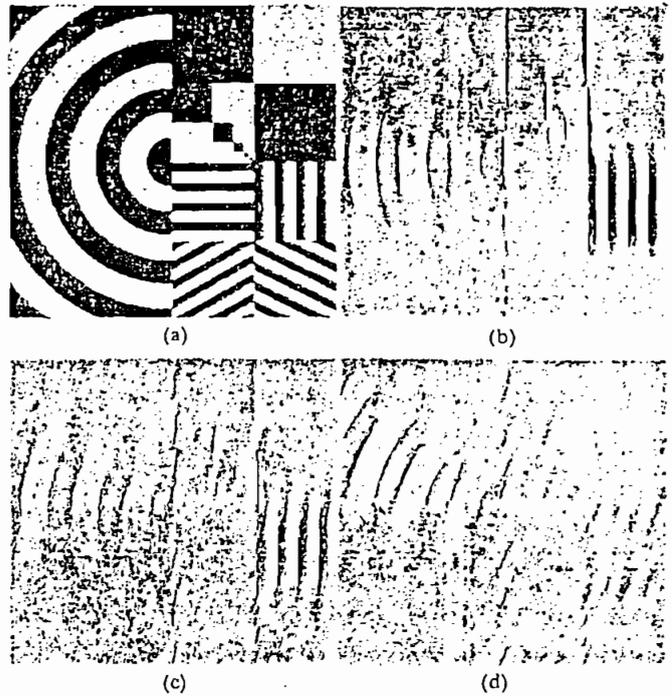


Fig. 33. (a) An original test image and (b)-(d) its first three directional components.

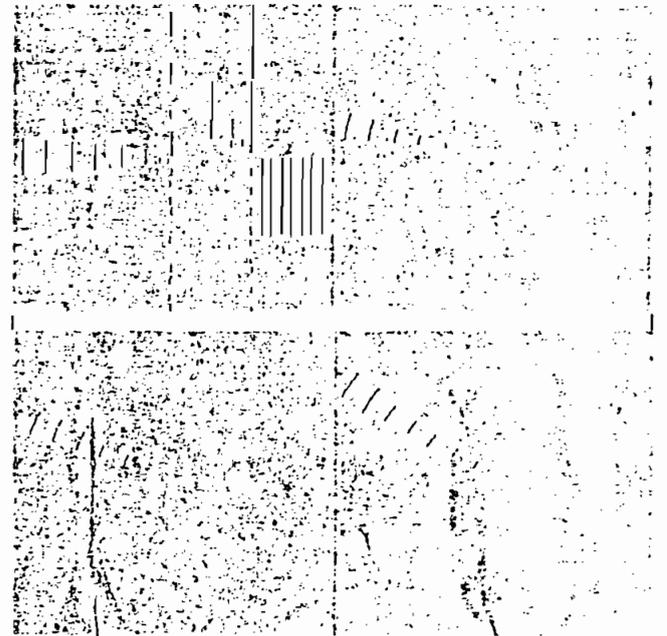


Fig. 34. Strong (dark) and weak (grey) edges in the first four directional images.

ANEXO 5.-

Image vector quatization.



Fig. 21. Original input image sequence of resolution (136, 200) with amplitude resolution of 8 bits [82].



Fig. 22. Processed image sequence at bit rate of $R = 1.6$ bits/pixel/frame with codebooks of size $CB1 = (285, 3)$ and $CB2 = (111, 6)$ with $NMSE = 9.2 - 9.5 \times 10^{-3}$ [82].



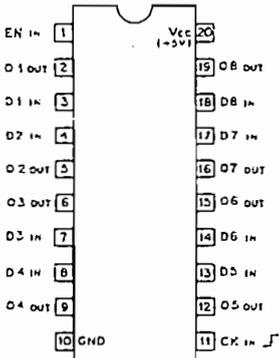
Fig. 23. Processed image sequence at bit rate of $R = 0.6$ bits/pixel/frame with codebooks of size $CB1 = (313, 4)$ and $CB2 = (203, 8)$ with $NMSE = 1.8 - 2.0 \times 10^{-3}$ [82].

ANEXO 6.-

Los circuitos integrados usados en la implementación son de características reguladas por el manual de la National.

A continuación se adjunta toda la posible información que se pudo recaudar de circuitos propios de Sony especiales para televisión.

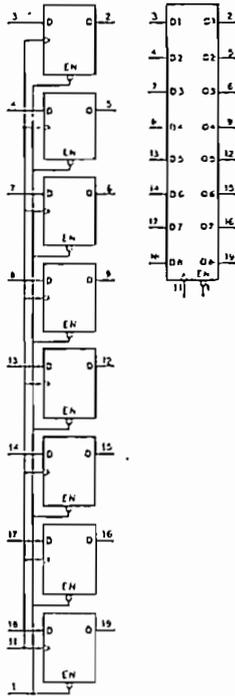
SN74LS377N (TI)
TTL D-TYPE FLIP-FLOP WITH ENABLE
— TOP VIEW —



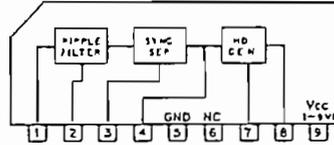
EACH FLIP-FLOP

INPUTS		OUT
EN	CK	Q
1	X	0
0	1	1
0	0	0
X	0	0

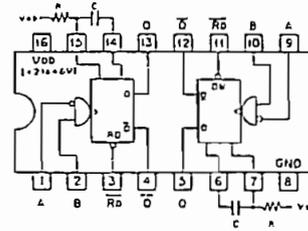
0, LOW LEVEL
1, HIGH LEVEL
X, DON'T CARE



TA73E7AP (TOSHIBA)
SYNC SEPARATOR/HD PULSE GENERATOR
— SIDE VIEW —



TC74HC123F (TOSHIBA) FLAT PACKAGE
CMOS DUAL RETRIGGERABLE MONOSTABLE MULTIVIBRATOR
— TOP VIEW —

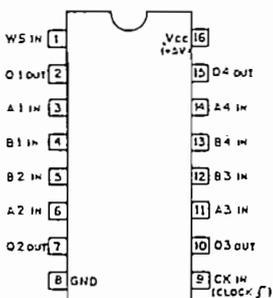


INPUT		OUTPUT
RD	A B	Q
0	X X	0
1	1 X	0
1	X 0	0
1	0 1	1
1	1 1	1

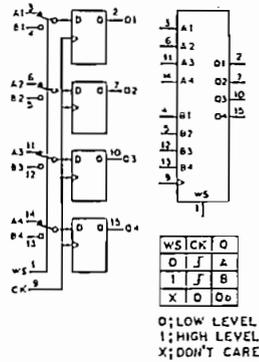
0; LOW LEVEL
1; HIGH LEVEL
X; DON'T CARE

OUTPUT PULSE WIDTH = 0.46CR

SN74LS399NS (TI) FLAT PACKAGE
TTL QUAD 2-INPUT MULTIPLEXER WITH STORAGE
— TOP VIEW —



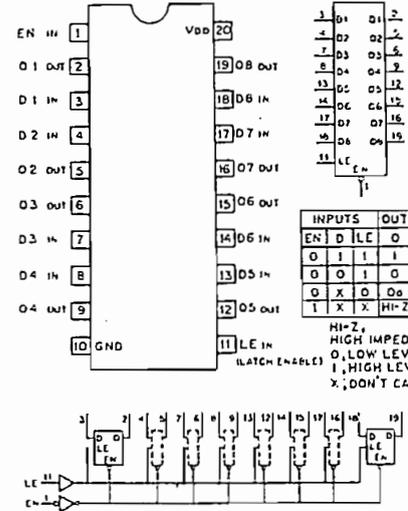
WS; WORD SELECT



INPUTS		OUT
A	B	Q
0	1	0
1	0	1
X	0	0

0; LOW LEVEL
1; HIGH LEVEL
X; DON'T CARE

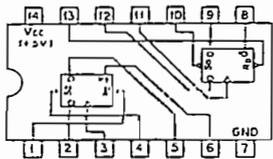
TC74HC373F (TOSHIBA) (V_{cc} = +2 to +6V) FLAT PACKAGE
CMOS 3 STATE OUTPUTS OCTAL LATCHES
— TOP VIEW —



INPUTS		OUT
EN	LE	Q
0	1	1
0	0	0
1	X	X

HI-Z, HIGH IMPEDANCE
0, LOW LEVEL
1, HIGH LEVEL
X; DON'T CARE

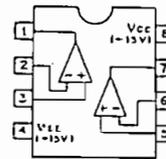
SN74LS74AH (TI)
SN74LS74ANS (TI) FLAT PACKAGE
TTL D-TYPE FLIP FLOP WITH DIRECT SET/RESET
— TOP VIEW —



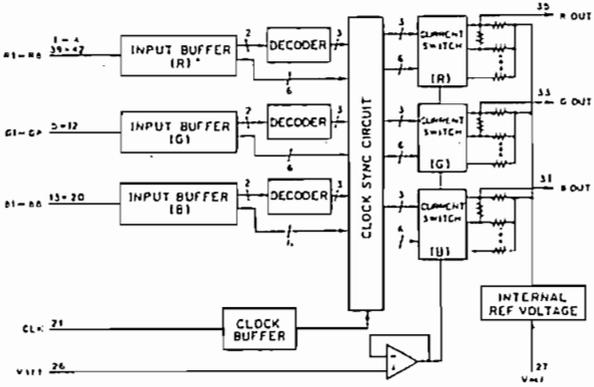
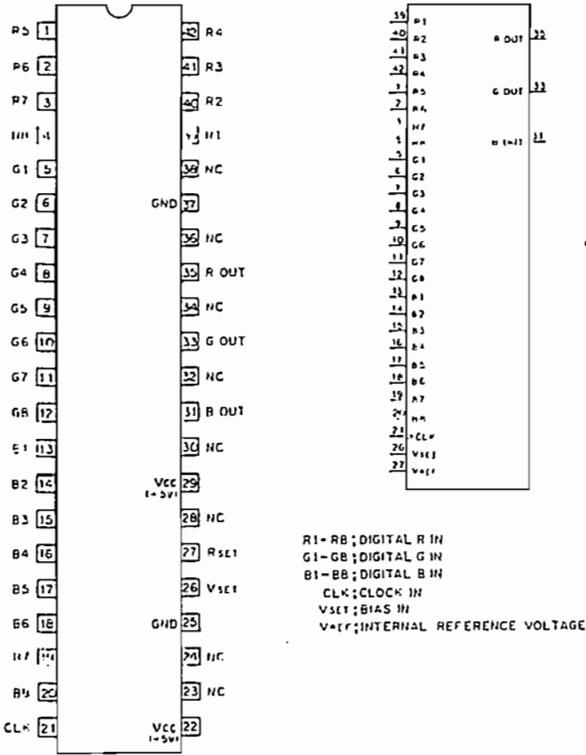
INPUTS		OUTPUTS	
S	R	Q	Qn
0	1	1	0
1	0	0	1
0	0	1*	1*
1	1	0	0
1	1	0	1
1	1	0	0

0; LOW LEVEL
1; HIGH LEVEL
X; DON'T CARE
1*; NONSTABLE

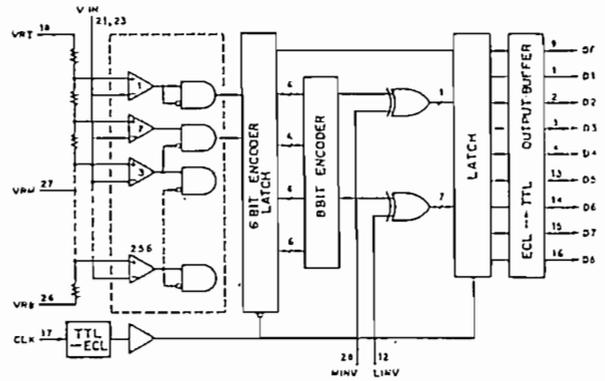
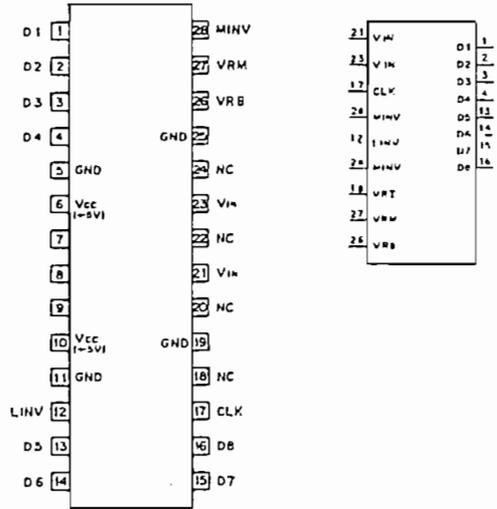
TL082CPS (TI) FLAT PACKAGE
OPERATIONAL AMPLIFIER
(J FET-INPUT)
— TOP VIEW — TL082CP



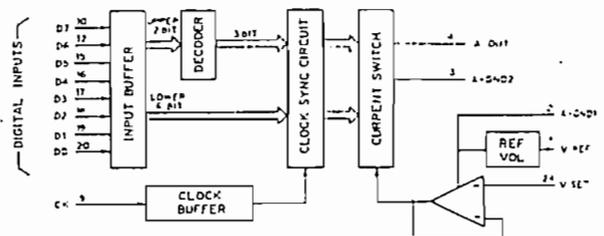
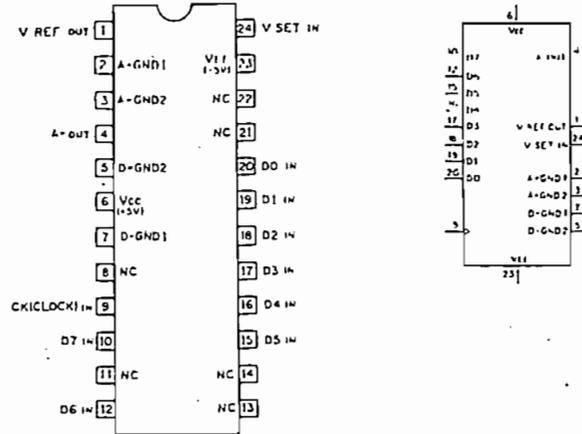
CXA1020F (SONY)
8 BIT 75MS/PS HIGH CHANNELS/D/A CONVERTER
TOP VIEW



CXA1096P (SONY)
8-BIT A/D CONVERTER



CXA1106M (SONY) FLAT PACKAGE
8-BIT D/A CONVERTER (TTL INPUT)
- TOP VIEW -



ECG Type	Description and Application	Collector To Base Volts BV _{CBO}	Collector To Emitter Volts BV _{CEO}	Base to Emitter Volts BV _{EBO}	Max. Collector Current I _C Amps	Max. Device Diss. P _D Watts	Freq. in MHz f _t	Current Gain h _{FE}	Package	
									Case	Fig. No.
ECG107	NPN-Si, UHF/VHF Amp, Osc, Mix, IF Amp	35	35	5	50 mA	.250 (T _A = 25°C)	800 min	70 typ	TO-92	T16
ECG108	NPN-Si, RF/IF/Video Amp, Osc, Mix, VHF/UHF	30	15	2	50 mA	.600 (T _A = 25°C)	800 min	20 min	TO-92	T16
ECG121 ECG121MP*	PNP-Ge, AF Pwr Output	65	45 (CER)	15	7.0	30	22 KHz #	80 typ	TO-3	T28
ECG123	NPN-Si, AF Preamp, Driver Video Amp, Sync Sep	60	30	5	.8	.800 (T _A = 25°C)	250	150 typ	TO-39	T6
ECG123A	NPN-Si, AF/RF Amp, Sw	75	40	6	.8	.500 (T _A = 25°C)	300	200 typ	TO-18	T2
ECG123AP	NPN-Si, AF/RF Amp, Driver (Compl to ECG159)	75	40	6	.6	.500 (T _A = 25°C)	300	200 typ	TO-92	T16
ECG124	NPN-Si, HV Audio Pwr Output	300	300	5	.150	20	30	100 typ	TO-66	T25
ECG126A	PNP-Ge, RF/IF Amp, Osc, Mix	15	15	3	50 mA	300 mW (T _A = 25°C)	250	40 typ	TO-18	T2
ECG127	PNP-Ge, Horiz & Vert Defl, Pwr Output	320	320 (CES)	2	10	40	1	15 min	TO-3	T28
ECG128	NPN-Si, AF Preamp, Driver, Output, Video Amp (Compl to ECG129)	120	80	7	1	1 (T _A = 25°C)	120	90 min	TO-39	T6
ECG128P	NPN-Si, Gen Purp Amp, Sw (Compl to ECG129P)	100	80	7	1	1	100	100 min	TO-237	T17
ECG129	PNP-Si, AF Preamp, Driver, Output, Video Amp (Compl to ECG128)	90	80	7	1	1 (T _A = 25°C)	120	90 min	TO-39	T6
ECG129MCP	Matched Compl Pair-Contains one each ECG128 (NPN) and ECG129 (PNP)									
ECG129P	PNP-Si, Gen Purp Amp, Sw (Compl to ECG128P)	80	80	7	1	1	150	100 min	TO-237	T17
ECG130 ECG130MP*	NPN-Si, AF Pwr Amp (Compl to ECG219)	100	60	7	15	115	.800	40 typ	TO-3	T28
ECG131 ECG131MP*	PNP-Ge, AF Pwr Output (Compl to ECG155)	32	20	10	3 peak	6 (T _C = 63°C)	1	110 typ	TC-9	T27
ECG152 ECG152MP*	NPN-Si, AF Pwr Output (Compl to ECG153)	60	60	5	7	50	10	60 typ	TO-220	T41
ECG153 ECG153MCP	PNP-Si, AF Pwr Output (Compl to ECG152) Matched Compl Pair-Contains one each ECG152 (NPN) and ECG153 (PNP)	60	60	5	7	50	10	60 typ	TO-220	T41
ECG154	NPN-Si, Video Output Amp	300	300	7	.5	1.0 (T _A = 25°C) 7.0 (T _C = 25°C)	40	60 typ	TO-39	T6
ECG155	NPN-Ge, AF Pwr Amp (Compl to ECG131)	32	20	10	3 peak	7.5	1	110 typ	TC-9	T27
ECG157	NPN-Si, HV AF Pwr Amp (Compl to ECG39)	300	300	3	.5	20.8	10	30 min	TO-126	T45
ECG158	PNP-Ge, AF Pwr Amp	32	32	10	1	1.6	1.5	90 typ	TO-1	T1
ECG159 ECG159MCP	PNP-Si, AF Preamp, Driver, Sw (Compl to ECG123AP) Matched Compl Pair-Contains one each ECG123AP (NPN) and ECG159 (PNP)	80	80	5	1	.600 (T _A = 25°C)	200	180 typ	TO-92	T16
ECG160	PNP-Ge, RF/IF Amp, Osc, Mix	30	20 (CES)	.5	10 mA	.200 (T _A = 25°C)	400	20 typ	TO-72	T4
ECG161	NPN-Si, Video IF Amp	45	45 (CES)	4.5	50 mA	.180 (T _A = 25°C)	800	60 typ	TO-72	T4

Notes: * MP Matched pair

Frequency at which common emitter current gain is 70.0% of low frequency gain

* When alternate packages are shown it indicates a change is in progress. Although only one package is available both packages will be shown as long as the obsolete package may be encountered in the field.

Package Outlines - See Page 1-76

ECG Type	Description and Application	To Base Volts BV _{CEO}	To Emitter Volts BV _{CEO}	Emitter Volts BV _{EBO}	Collector Current I _C Amps	Device Diss. P _D Watts	In MHz f _t	Current Gain h _{FE}	Case	Fig. No.
ECG290A ECG290AMP	PNP-Si, AF Pwr Amp (Compl to ECG289A) Matched Compl Pair-Contains one each ECG289A (NPN) and ECG290A (PNP)	100	80	5	.5	.500 (T _A = 25°C)	120	100 min	TO-92	T16
ECG291	NPN-Si, Pwr Amp, Sw (Compl to ECG292)	130	120	5	4	40	4 min	75 typ	TO-220	T41
ECG292 ECG292MCP	PNP-Si, Pwr Amp, Sw (Compl to ECG291) Matched Compl Pair-Contains one each ECG291 (NPN) and ECG292 (PNP)	130	120	5	4	40	4 min	75 typ	TO-220	T41
ECG293 ECG293MP*	NPN-Si, AF Pwr Amp (Compl to ECG294)	60	50	5	1	1 (Heat Sink) .75 (T _A = 25°C)	200	120 min	T-16	T20
ECG294	PNP-Si, AF Pwr Amp (Compl to ECG293)	60	50	5	1	1 (Heat Sink) .75 (T _A = 25°C)	200	120 min	T-16	T20
ECG295	NPN-Si, RF Output/Driver	75	40	5	1	5	150 min	20 min	TO-126	T45
ECG297 ECG297MP*	NPN-Si, AF Driver, Pwr Amp (Compl to ECG298)	80	80	5	1 peak	.75 (T _A = 25°C)	120	130 min	T-16	T20
ECG298	PNP-Si, AF Driver/Pwr Amp (Compl to ECG297)	80	80	5	1 peak	.75 (T _A = 25°C)	120	130 min	T-16	T20
ECG300 ECG300MP*	NPN-Si, AF Pwr Output (Compl to ECG307)	50	40	5	1.5	7	70	90 min	TO-202	T38
ECG302	NPN-Si, RF Driver/Pwr Amp	100	50	6	1.5 peak	8	80	200 min	TO-202J	T37
ECG304A	Transistor Kit See Page 1-53	---	---	---	---	---	---	---	---	---
ECG306	NPN-Si, RF Driver/Pwr Amp	100	50	6	1.5 peak	8	80	200 min	TO-202J	T37
ECG307	PNP-Si, AF Pwr Output (Compl to ECG300)	50	40	5	1.5	7	70	90 min	TO-202	T38
ECG311	NPN-Si, VHF/UHF Osc, Amp, Driver	55	30	3.5	.4	5	800 min	25 min	TO-39	T6
ECG312	See FET Selector Guide Page 1-56	---	---	---	---	---	---	---	---	---
ECG313	NPN-Si, VHF Tuner, RF Amp	30	30	4	20 mA	.15 (T _A = 25°C)	400 min	25 min	RF-15	T49
ECG315	NPN-Si, RF Driver	100	50	6	1	.75 (T _A = 25°C)	80	200 typ	SC-51	T19
ECG318	NPN-Si, Lo Noise UHF Amp (NF 4.5 dB max, 450 MHz)	30	15	3.5	50 mA	.200 (T _A = 25°C)	1400	25 min	TO-72	T4
ECG317	NPN-Si, RF Pwr Output (P _O 70 W, 30 MHz)	36	36 (CES)	4	15	220	---	10 min	RF-50F	T58
ECG318	NPN-Si, RF Pwr Output (P _O 50 W, 30 MHz)	36	36 (CES)	4	6	80	---	5 min	RF-50F	T58
ECG319P	NPN-Si, AGC Controlled IF Amp (Includes Metal Shield)	40	30	4	50 mA	.5 (T _A = 25°C)	300 min	35 typ	TO-92	T16
ECG320	NPN-Si, RF Pwr Output (P _O 40 W, 175 MHz)	36	36 (CES)	4	6	80	200 min	5 min	RF-50SS	T55
ECG320F	NPN-Si, RF Pwr Out (P _O 40 W, 175 MHz)	36	18	4	6	80	200 min	5 min	RF-38F	T57
ECG321	NPN-Si, Horiz Defl	1600	700	5	1.5	15 (T _C = 25°C)	---	1.5 min	TO-66	T25
ECG323	PNP-Si, AF Pwr Amp (Compl to ECG324)	120	100	5	2	1 (T _A = 25°C)	30	50 min	TO-39	T6

Notes: * MP - Matched pair

† Frequency at which common emitter current gain is 20 dB of low frequency gain

• When alternate packages are shown it indicates a change is in progress. Although only one package is available both packages will be shown as long as the obsolete package may be encountered in the field.

Package Outline - See Page 1-76

ECG Type	Description and Application	Collector To Base Volts BV _{CBO}	Collector To Emitter Volts BV _{CEO}	Base to Emitter Volts BV _{EBO}	Max. Collector Current I _C Amps	Max. Device Diss. P _D Watts	Freq. in MHz f _t	Current Gain h _{FE}	Package	
									Case	Fig. No.
ECG260	PNP-Si, Darlington Pwr Amp (Compl to ECG259)	100	100	5	8	75	---	2500 typ	TO-127	T46
ECG261	NPN-Si, Darlington Pwr Amp (Compl to ECG262)	100	100	5	8	65	---	1000 min	TO-220	T41
ECG262	PNP-Si, Darlington Pwr Amp (Compl to ECG261)	100	100	5	8	65	---	1000 min	TO-220	T41
ECG263	NPN-Si, Darlington Pwr Amp (Compl to ECG264)	100	100	5	10	65	---	1000 min	TO-220	T41
ECG264	PNP-Si, Darlington Pwr Amp (Compl to ECG263)	100	100	5	10	65	---	1000 min	TO-220	T41
ECG265	NPN-Si, Darlington Pwr Amp, Switch	50	50	13	.5	6.25	---	10000 min	TO-202	T38
ECG266	NPN-Si, Darlington Pwr Amp, Switch	50	50	13	.5	6.25	---	40000 min	TO-202	T38
ECG267	NPN-Si, Darlington Pwr Amp, Switch	30	30	13	.5	6.25	---	90000 min	TO-202	T38
ECG268	NPN-Si, Darlington Pwr Amp (Compl to ECG269)	50	50	13	2	10	---	1000 min	TO-202	T38
ECG269	PNP-Si, Darlington Pwr Amp (Compl to ECG268)	50	50	13	2	10	---	1000 min	TO-202	T38
ECG270	NPN-Si, Darlington Pwr Amp, Switch (Compl to ECG271)	100	100	5	10	125	---	1000 min	TO-3P (TO-218)	T48
ECG271	PNP-Si, Darlington Pwr Amp, Switch (Compl to ECG270)	100	100	5	10	125	---	1000 min	TO-3P (TO-218)	T48
ECG272	NPN-Si, Darlington Pwr Amp, Switch (Compl to ECG273)	50	40	12	2	10	---	25000 min	TO-202N	T36
ECG273	PNP-Si, Darlington Pwr Amp, Switch (Compl to ECG272)	50	40	12	2	10	---	25000 min	TO-202N	T36
ECG274	NPN-Si, Darlington Pwr Amp, Switch (Compl to ECG275)	80	80	5	4	50	---	3000 typ	TO-66	T25
ECG275	PNP-Si, Darlington Pwr Amp, Switch (Compl to ECG274)	80	80	5	4	50	---	3000 typ	TO-66	T25
ECG278	NPN-Si, Broad Band RF Amp	40	30	4	.4	3.5	1200 min	30 min	TO-39	T6
ECG280 ECG280MP*	NPN-Si, AF Pwr Amp (Compl to ECG281)	140	140	5	12	100	6	70 min	TO-3	T28
ECG281 ECG281MCP	PNP-Si, AF Pwr Amp (Compl to ECG280) Matched Compl Pair-Contains one each ECG280 (NPN) and ECG281 (PNP)	140	140	5	12	100	6	70 min	TO-3	T28
ECG283	NPN-Si, HV-Hi Current Sw, Horiz Output	800	325	8	10	100	6	15 min	TO-3	T28
ECG284 ECG284MP*	NPN-Si, AF Pwr Amp (Compl to ECG285)	180	180	5	16	150	6	70 min	TO-3	T28
ECG285 ECG285MCP	PNP-Si, AF Pwr Amp (Compl to ECG284) Matched Compl Pair-Contains one each ECG284 (NPN) and ECG285 (PNP)	180	180	5	16	150	6	70 min	TO-3	T28
ECG286	NPN-Si, Pwr Amp, Sw, Horiz Output	300	250	6	2	25	8	100 min	TO-66	T25
ECG287	NPN-Si, HV Gen Purp Amp (Compl to ECG288)	300	300	6	.5	.625 (T _A = 25°C)	50	40 min	TO-92	T16
ECG288	PNP-Si, HV Gen Purp Amp (Compl to ECG287)	300	300	5	.5	.625 (T _A = 25°C)	50	40 min	TO-92	T16
ECG289A ECG289AMP*	NPN-Si, AF Pwr Amp (Compl to ECG290A)	100	80	5	.5	.500 (T _A = 25°C)	120	100 min	TO-92	T16

Notes: * MP - Matched pair

Package Outlines - See Page 1-76

f Frequency at which common emitter current gain is 70.0% of low frequency gain

* When alternate packages are shown it indicates a change is in progress. Although only one package is available both packages will be shown as long as the obsolete package may be encountered in the field.

NTE TYPE NO.	Polarity and Material	Description and Application	Case Style	Diag No.	Breakdown Voltage	Cutoff Voltage	Drain Current	Drain Current	Resistance	Cap	Reverse	Transcon-	Power
					V _{GS} Gate To Source Min	V _{GS} Gate To Source Max (off)	Zero-Gate Min	Max	Drain to Source ohms Max (on)	Input Max pf	Transfer Capacitance Max pf	ductance μ mhos typical	Dissipation Max mW
					BV _{GSS}	V _{GS}	I _{DSS}	I _D	r _{DSS}	C _{iss}	C _{rss}	g _m	P _D
132	JFET N-CH	VHF Amp/Mix, NF 4db Max at 400 MHz	TO106	29b	25	6	8 - 20	—	—	5	1.2	4,500	310
133	JFET N-CH	Gen Purp AF Amp/Sw	TO106	29b	30	6	5 - 15	—	—	6	2	4,000	300
220	MOSFET N-CH	VHF Amp/Mix, NF 5db Max at 200 MHz	TO72	31c	20	8	5 - 25	—	—	7	.35	7,500	330
221	Dual Gate MOSFET N-CH	VHF Amp/Mix, NF 5db Max at 200 MHz	TO72	31d	20	6	18 typ	—	—	5.5	.03	15,000	400
222	Dual Gate MOSFET N-CH	VHF Amp/Mix, NF 6db Max at 200 MHz Gate Protected	TO72	31d	20	4	5 - 35	—	—	6 typ	.03	12,000	330
312	JFET N-CH	VHF Amp/Mix, NF 4db Max at 400 MHz	TO92	9d	30	6	5 - 15	—	—	4.5	1	5,500	360
326	JFET P-CH	Gen Purp AF Amp, NF 2.5 db Max at 100 Hz	TO92	9o	60	7.5	2 - 9	—	—	7	2	3,000	310
451	JFET N-CH	UHF/VHF Amp, NF 4db at 400 MHz	TO92	9e	25	4	4 - 10	—	—	5	1	4,000	310
452	JFET N-CH	VHF Amp/Mix NF 4dB at 400 MHz	TO72	31e	30	6	5 - 15	—	—	4	8	5,500	300
453	JFET N-CH	FM Tuner/VHF Amp, NF 2.0dB at 100 MHz	TO98M TO92	36 9e	18	3 typ	12 - 24	—	—	—	65	7,000	200
454	Dual Gate MOSFET N-CH	UHF/VHF Preamp, NF 4.5dB at 200 MHz Gate Protected	TO72	31d	25 (BVDSX)	5	6 - 30	—	—	3.3 typ	.03	15,000	360
455	Dual Gate MOSFET N-CH	TV UHF/RF Amp 900 MHz Range, Gate Protected	—	12	20 (BVDSX)	G ₁ = -2 G ₂ = -.7	0.5 - 8	25	—	2.5	.03	22,000	200
456	JFET N-CH	Gen Purp Amp/Sw NF 2.5 db @ 100 KHz	TO72	31c	30	6	2 - 6	15	400	6	2	3,500	300
457	JFET N-CH	Gen Purp Amp/Sw	TO92	9e	25	6	1 - 5	—	—	7	3	3,000	310
458	JFET N-CH	Gen Purp Low Noise NF 3.0 db @ 100 KHz	TO92	9f	50	1.5	0.5 - 12	20	—	13	2.6 typ	12,000	250
459	JFET N-CH	AF Amp/Chopper/Sw	TO72	31e	50	6	2 - 10	10 mA	—	6	3	4,500	300
460	JFET P-CH	AF Amp, AC Input Impedance 5M @ 1 KC	TO72	31f	20	6	2 - 6	10	800	20	—	2,500	300
461	Dual Matched Pr JFET N-CH	DC Amp/Sampler/Chopper (.95 ratio)	TO71	44	50	4.5	0.5-8	30	—	6	2	3,500	250(each) 400(total)
462	MOSFET N-CH	Designed for low pwr app in audio freq range	TO18	28b	20 (BVDSX)	7	2-6	20	—	8	.8	2,300	300
464	MOSFET P-CH	Enhancement mode, switching app (Compl to NTE465)	TO72	31g	25	5	10 nA Max	30	600	5	1.3	1000 min	300
465	MOSFET N-CH	Enhancement mode, switching app (Compl to NTE464)	TO72	31L	25	5	10 nA Max	30	300	5	1.3	1000 min	300
466	JFET N-CH	Chopper/Sw	TO18	28c	40	10	50 mA Min	250 pA	25	18	—	—	360
467	JFET N-CH	Chopper/High Speed Switching	TO92	9e	30	12	50 mA Min	1.0 nA	30	10	4	—	310