UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

DIVISION DE ESTUDIOS DE POSGRADO DE LA FACULTAD DE INGENIERIA

Compresión de Video Basada en Regiones

T E S I S

QUE PARA OBTENER EL GRADO DE:
MAESTRO EN INGENIERIA ELECTRICA
PRESENTA

BENJAMÍN VALERA OROZCO

DIRECTOR DE TESIS: DR. VICTOR GARCIA GARDUÑO

SINODALES

DR. BORIS ESCALANTE RAMIREZ
DR. FRANCISCO GARCIA UGALDE
DR. JORGE LIRA CHAVEZ
DR. MIGUEL MOCTEZUMA FLORES

MEXICO, D.F. Mayo del 2000

AGRADECIMIENTOS

Al Centro de Instrumentos de la UNAM, el lugar donde trabajo y he encontrado un enorme apoyo.

Al Dr. Víctor García Garduño por su paciencia y amistad.

A mis compañeros de trabajo de la Sección de Metrología del CI UNAM, por su amistad.

Al Ing. Gerardo Ruiz Botello, por haberme apoyado en todo momento.

Contenido

Introducción

Los servicios actuales que requieren compresión de imágenes y vídeo con bajas tasas de bit siguen utilizando en gran medida líneas de transmisión en banda base con anchos de banda estrechos. Tal es el caso, por ejemplo, de los servicios de comunicaciones móviles, videoteléfono e internet, que en mayor medida llegan al público mediante el uso tradicional de la red pública telefónica conmutada. Ante ésta situación prevaleciente y con pocas perspectivas de contar en el corto plazo con una red pública con mayor ancho de banda, (fibra óptica) la codificación y transmisión digital de imágenes fijas y secuencias de vídeo con bajas tasas de bit sigue teniendo un amplio campo de trabajo.

El enfoque actual que está siendo impulsado para tomarse en consideración en la norma MPEG4 [10] es orientar las técnicas de procesamiento a una técnica basada en regiones u objetos en lugar de la utilizada en MPEG2 orientada a pixeles. Se ha comprobado que con algunas restricciones y bajo aplicaciones particulares [11] se pueden lograr tasas de transmisión cercanas a 64 kb/s.

En éste sentido, y debido a la importancia vigente de las técnicas de compresión de vídeo con bajas tasas de bit, el propósito de éste trabajo es exponer algunas consideraciones prácticas en la implementación de un algoritmo para compresión basado en regiones. Lo expuesto aquí está inspirado en gran parte por los trabajos previos en [29] y por un interés particular de atender la compresión de video con resultados lo más cercano posible a lo que se realiza actualmente.

Un tema sobresaliente abordado en éste trabajo es la morfología matemática [12, 28], en particular los filtros morfológicos por reconstrucción que son muy atractivos como parte de un procesamiento previo de la imagen. El objetivo es acondicionar la imagen para las etapas posteriores de procesamiento de manera que se reduzca su carga computacional. Bajo éste esquema, las decisiones acerca de la homogeneidad y similitud durante el procesamiento son más marcadas, repercutiendo en una apariencia mayormente enfocada a la percepción de la visión humana, eliminando la sobrecarga computacional ocasionada por las pequeñas áreas de alto contraste y menor relevancia perceptual. Por lo tanto, las transformaciones por reconstrucción no sólo eliminan detalles con alto contraste localizados en pequeñas áreas, también

preservan los contornos de la imagen [20]. En otras palabras, los filtros por reconstrucción no eliminan las componentes de frecuencia que marcan los contornos de una imagen como lo hacen algunos filtros lineales. Tampoco alteran el patrón básico de la imagen inspeccionada como lo hacen los filtros de mediana o la apertura y cierre morfológicos. Estos remueven y fusionan zonas planas al conectar áreas equivalentes al elemento estructurante utilizado.

Las notables ventajas que ofrecen los filtros por reconstrucción como parte del acondicionamiento y preprocesamiento de imágenes han dado la pauta para elaborar una sólida base teórica [23] implementaciones robustas y eficientes [28] y han encontrado gran aceptación en aplicaciones concretas orientadas a la codificación con bajas tasas de bit [29].

El salto tecnológico de los algoritmos de codificación de vídeo a bajas tasas de bit está vinculados al desarrollo de técnicas de segmentación eficientes. En éste sentido, los más importantes grupos de trabajo reconocen el desarrollo de una nueva generación de técnicas de codificación de vídeo no convencionales, basada en regiones u objetos (MPEG4) en lugar de las tradicionales basadas en pixeles (MPEG2) [10]. Con tal situación, la tendencia es codificar video en forma más eficiente, proporcionando mayor prioridad a las facultades de percepción humana, es decir encontrar regiones u objetos y aislarlos de la escena. Cuando un ser humano observa la escena, el procesamiento que se lleva a cabo en el sistema visual esencialmente segmenta la escena exclusivamente para el o ella. Esto se realiza de una manera espontanea y eficiente para el observador en escenas no muy complejas. Sin embargo, en procesamiento digital de imágenes el asunto no es tan trivial.

La segmentación de imágenes es, por lo tanto, un paso fundamental en la codificación de vídeo, visión por computadora y reconocimiento de patrones. Este trabajo presenta un método de segmentación espacial de imágenes basado en simplificación, división y fusión [6]. El algoritmo empleado primero simplifica las imágenes utilizando filtros morfológicos por reconstrucción. Las pequeñas zonas planas de la imagen original se fusionan con otras similares en tamaño y similitud. Posteriormente la imagen se divide de acuerdo al algoritmo del árbol cuádruple, en un conjunto de regiones arbitrarias disjuntas pero que siguen el patrón básico de discontinuidades en la intensidad. A continuación en la imagen dividida, que en éste momento está sobresegmentada, se fusionan regiones vecinas progresivamente al seleccionar una región pivote en forma aleatoria y examinar si sus vecinos cumplen con un criterio de similitud para su fusión. El progreso presentado en éste trabajo está en la capacidad de fusión de regiones pivote, que se adapta en forma significativa al fenómeno de percepción visual humano, ya que la región pivote tiende a adaptar su parámetro descriptor a la escena. Por lo tanto, regiones vecinas que en un primer intento no fueron fusionadas pueden fusionarse posteriormente a la vez que la región pivote se va adaptando a una región u objeto particular de la escena. Posteriormente el proceso de fusión deja un remanente de regiones pequeñas, con menor importancia perceptual,

que deben eliminarse si se desea una segmentación más completa. En la etapa final, la segmentación se completa al controlar el número final de regiones.

Por otra parte, el análisis y estimación de movimiento para la segmentación espacio temporal son técnicas muy importantes para la codificación eficiente de video que se obtiene de una cámara monocular. El movimiento 2D es una proyección del movimiento 3D en escenas reales. Entonces, el modelo de movimiento 2D describe la relación entre regiones de imágenes sucesivas, y proporciona un conjunto reducido de parámetros que permiten la codificación a bajas tasas de bit. En nuestro trabajo ensayamos con un modelo simplificado de 4 parámetros orientado a regiones, que resulta tener un buen compromiso entre representatividad y economía. Los parámetros del modelo son obtenidos mediante la minimización de estimadores de máxima probabilidad (estimadores M) robusto y cuadrático, empleando el método del gradiente.

Finalmente, se realiza la codificación de la segmentación espacio temporal, con el objeto de obtener altas relaciones de compresión. La segmentación espacio temporal inicial, representada por las fronteras y el conjunto de parámetros de movimiento en t=1, debe transmitirse enteramente así como la primera imagen en t=0 codificada en JPEG, por ejemplo. Para la transmisión de imágenes sucesivas, solo es suficiente transmitir la codificación de la imagen de diferencia desplazada (DFD), es decir el error entre la imagen en $t=t_0$ y la imagen reconstruida a partir de los parámetros de movimiento en $t=t_0+\Delta t$. El esquema completo de compresión se ilustra en la figura 1.

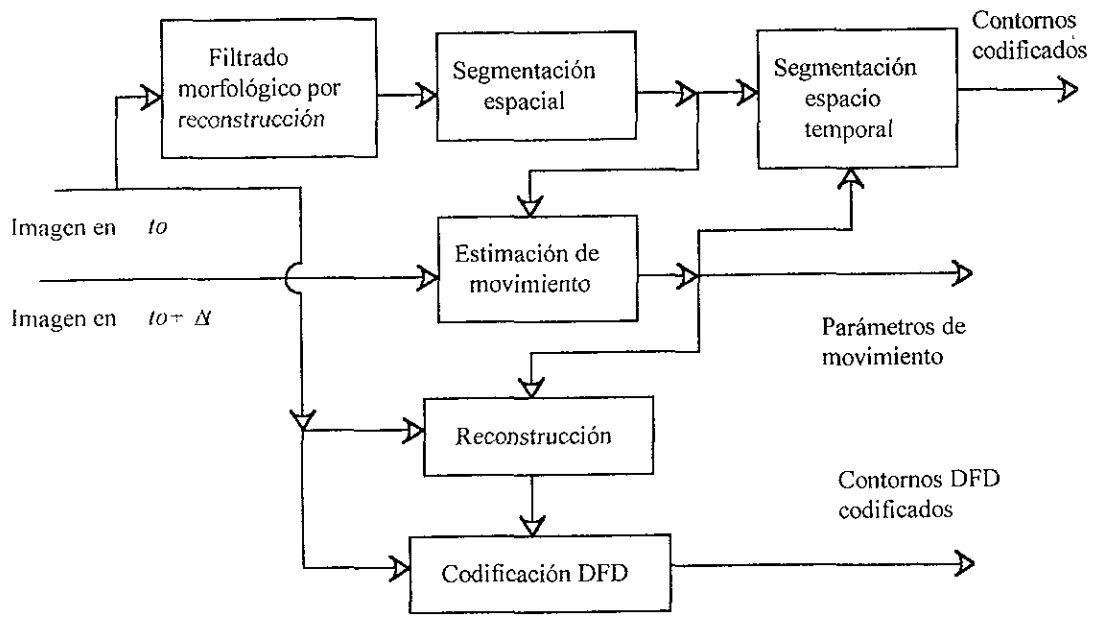


Figura 1. Esquema de compresión.

El objetivo principal de éste trabajo es el estudio e implementación en lenguaje de programación C de un algoritmo para la compresión digital de video. Teniendo en cuenta la gran diversidad de técnicas para el procesamiento de imágenes, nuestro interés está en la utilización de

herramientas de morfología matemática. La organización de la tesis se describe a continuación.

El capítulo uno presenta una revisión teórica de las técnicas más promisorias para la codificación de video a bajas tasas de bit. Dos tipos de técnicas potenciales son abordadas: codificación basada en forma de onda y codificación basada en modelado.

En el capítulo dos revisamos teóricamente las técnicas más comunes para el filtrado espacial de imágenes, enfocando el estudio a un preprocesamiento para las etapas posteriores. Enfatizamos en los filtros morfológicos por reconstrucción que no eliminan contornos útiles en las etapas de segmentación. Al mismo tiempo, presentamos resultados de filtros implementados en lenguaje C.

En el capitulo tres presentamos la teoría básica para los dos tipos de segmentación espacial existentes: basada en bordes y basada en regiones. Nuestra elección para la implementación en lenguaje C es por la segunda, debido al enfoque actual hacia regiones de la norma MPEG4.

El capítulo cuatro comienza con un análisis del modelo de movimiento 3D proyectado en el plano imagen. Posteriormente se discuten las tres técnicas generales para la estimación de los parámetros del modelo: técnicas de regresión, de correlación, y de lisado explícito. Enfocamos nuestro esfuerzo de programación en la estimación robusta de movimiento, ya que se ha comprobado su inmunidad a las perturbaciones. En la parte final simplificamos la segmentación espacial obtenida anteriormente al fusionar regiones espaciales con movimiento similar.

En el capítulo cinco revisamos algunas técnicas de codificación orientadas a regiones, de manera que las relaciones de compresión sean eficientes. También presentamos ejemplos de implementación para la codificación, principalmente para la imagen de error.

El capitulo seis presenta conclusiones y resultados ilustrativos de la técnica empleada, como lo son las gráficas de relación señal a ruido y tasa de bit contra número de secuencia en el video. De igual forma discutimos limitaciones así como trabajo a futuro.

En forma adicional, presentamos dos apéndices breves en donde se resumen algunas técnicas matemáticas para la minimización de funciones multidimensionales así como una descripción de una posible aplicación de las técnicas de análisis de movimiento para la visión estéreo.

Capitulo 1

Compresión de video

1.1. Introducción general

Este capítulo presenta una revisión de las técnicas más promisorias para la codificación de video a bajas tasas de bit, inferiores a 64 kb/s. La codificación de secuencias de imágenes a tales tasas de bit será una técnica crucial en los servicios visuales venideros, verbigracia transmisión y almacenamiento de información visual. Una aplicación típica es transmitir escenas en movimiento de videoteléfonos a través de las líneas telefónicas analógicas existentes. En el presente capítulo, exploramos dos tipos de técnicas potenciales: compresión basada en forma de onda y compresión basada en modelado.

En la compresión basada en forma de onda una imagen es tratada como una señal con forma de onda 2D o un segmento de una secuencia de imágenes como una señal con forma de onda 3D, explotando sus características estadísticas o determinísticas inherentes. Algunas técnicas para la codificación de imágenes, tales como la *codificación por transformación*, *codificación sub-banda/ondeletas*, *codificación VQ* y *codificación fractal*, pueden incluirse en este grupo. En compresión basada en modelado, la imagen de entrada es vista como una proyección 2D de la escena física real 3D. La codificación es implementada al modelar primero la escena 3D, y entonces se extraen los parámetros del modelo en el codificador. Finalmente se sintetiza la imagen en el decodificador a partir de los parámetros cuantizados.

1.2. Compresión basada en forma de onda

En compresión basada en forma de onda, la codificación se desarrolla directamente sobre la distribución de intensidades bidimensional. El problema básico que se enfrenta es lograr la mínima distorsión de forma de onda posible para una tasa de bit, o equivalentemente, lograr un nivel aceptable de distorsión en la forma de onda con la menor tasa de bit posible.

Una secuencia de imágenes es información 3D fluyendo que provee una gran cantidad de información espacial temporal y espectral (usando transformaciones). Sin embargo, existe mucha redundancia inherente, especialmente para escenas restringidas (videoconferencia, videoteléfono, etc.). Por lo tanto los esquemas de compresión deben hacer uso de tal redundancia, especialmente en características como: incorporar análisis de movimiento mediante la representación de secuencias de imágenes a través de un conjunto reducido de parámetros de movimiento, y orientar el enfoque a regiones en lugar de pixeles.

1.2.1. Análisis de movimiento

En compresión a bajas tasas de bit, necesitamos tener un buen conocimiento del comportamiento del movimiento. El compromiso entre cada vez más bajas tasas de bit está a la par de un análisis de movimiento cada vez más complejo. En éste trabajo, las cuatro técnicas para el análisis de movimiento que abordamos son: *modelos de movimiento*, *estimación de movimiento* *compensación de movimiento* y *codificación de los parámetros de movimiento.*

Modelos de movimiento

El movimiento 3D relativo entre la cámara y objetos resulta en un cambio temporal en el plano imagen 2D. Tal cambio temporal puede modelarse por *modelos de proyección.* Existen dos tipos generales de proyecciones: ortogonal y perspectiva. De ésta manera, los cambios temporales son descritos por las siguientes ecuaciones [11]:

$$[\,x'\ y'\ x\ \ y\ \ 1\,][e1\ e2\ e3\ e4\ e5]^T = 0 \quad \text{proyección ortogonal} \qquad (1.1)$$

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} \begin{bmatrix} e1 & e2 & e3 \\ e4 & e5 & e6 \\ e7 & e8 & e9 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0 \quad \text{proyección perspectiva} \qquad (1.2)$$

en donde ei es el parámetro de movimiento, $(x\ \ y)$ y $(x'\ y')$ son posiciones antes y después del movimiento.

Para aplicaciones restringidas, dos grupos de modelos de movimiento son de especial interés: *movimiento de cámara* y *movimiento de objeto.* Con el movimiento de cámara, el movimiento global está limitado para ciertas aplicaciones como videoteléfono. Sólo algunos movimientos globales son permitidos. Con éstas restricciones en el movimiento, se pueden plantear modelos simplificados. Por el contrario, con el movimiento de objeto, el análisis es más complicado ya que los objetos pueden tener desplazamientos más diversos, ya sean rígidos o no rígidos. Las tablas 1.1 y 1.2 resumen los principales modelos de movimiento.

Pan	Zoom	Vibración
$x' = x + a0$ $y' = y + b0$	$x' = a1\ x$ $y' = b1\ y$	$x' = a0 + a1\ x + a2\ y$ $y' = b0 + b1\ x + b2\ y$
Causado por la traslación de la cámara	Causado por el acercamiento o alejamiento de la cámara	Causado por el movimiento 3D de la cámara

Tabla 1.1. Modelos de cámara simplificados.

Ortogonal	Perspectiva
$x' = a0 + a1\ x + a2\ y$ $y' = b0 + b1\ x + b2\ y$	$x' = \frac{a0 + a1x + a2y}{1 + a3 + a4y}$ $y' = \frac{b0 + b1x + b2y}{1 + b3 + b4y}$
Movimiento sobre parches planos bajo proyección ortogonal	Movimiento sobre parches planos bajo proyección perspectiva

Tabla 1.2. Modelos de objeto simplificados.

Estimación de movimiento

Una vez que el movimiento ha sido modelado, el problema de estimación de movimiento se convierte en un problema de estimación de parámetros. La estrategia en la estimación de movimiento se ilustra en la figura 1.1.

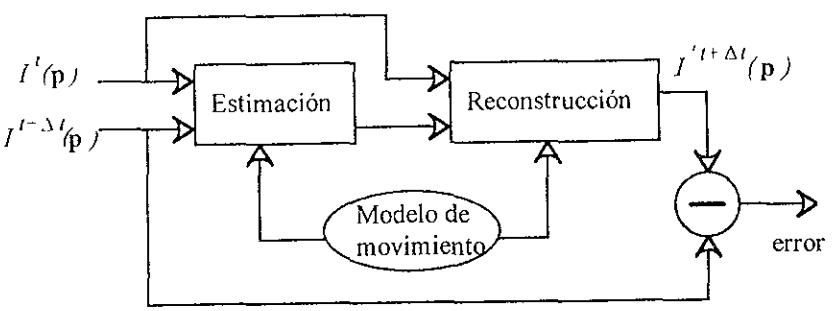


Figura 1.1. Estimación de movimiento.

en donde $\mathbf{p}=(x\ y)$ son las coordenadas del plano imagen, $I^t(\mathbf{p})$ es la imagen en el tiempo t, $I^{t-\Delta t}(\mathbf{p})$ es la imagen en el tiempo $t+\Delta t$, e $I'^{t+\Delta t}(\mathbf{p})$ es la imagen estimada en el tiempo $t+\Delta t$.

En la figura 1.1, se ilustra un esquema típico de minimización que proporciona un método eficiente de estimación al minimizar el error de reconstrucción. En la etapa de estimación se extraen los parámetros de movimiento al encontrar el mejor emparejamiento (best Match) entre las imágenes en t y $t + \Delta t$. La señal de error es utilizada como criterio para medir

la calidad de la estimación. Esta puede ser introducida en un lazo de realimentación para realizar cálculos iterativos. No obstante, en el uso práctico de tal estrategia, se deben tomar en cuenta algunos aspectos finos:

a) Definición del criterio de error. Para recuperar un estimado del flujo óptico (gradiente temporal) en un intervalo de video, simplemente se puede minimizar la diferencia de imágenes desplazada (DFD):

$$DFD=\rho(I^{t}(x\ y)- I^{t+\Delta t}(x'\ y'))$$

en donde x' y y' pertenecen a alguno de los modelos de movimiento en las tablas 1.1 o 1.2, ρ es normalmente un estimador de máxima probabilidad (estimador M) que puede ser el estándar de mínimos cuadrados $\rho(x) = x^2$, o alguno de los estimadores robustos [2] discutidos en el capítulo cuatro.

b) Estimación de movimiento local a una región. La estrategia principal para manejar este problema es encontrando regiones homogéneas en intensidad en donde todos los parámetros de movimiento pertenezcan a un modelo unificado y posteriormente estimar los parámetros. En éste sentido, es útil emplear técnicas de segmentación espacial orientadas a regiones, que pueden ser de tres tipos básicos: *división en regiones fijas*, en donde la imagen de entrada es dividida en regiones disjuntas de acuerdo a una estructura de rejilla predeterminada. Las rejillas más comunes son parches de rectángulos o triángulos; *división en regiones adaptables*, en donde la imagen de entrada se divide en forma adaptable de acuerdo a un criterio de homogeneidad o similitud. En éste caso la técnica más común es la división por árbol cuádruple (quadtree), discutida en el capítulo tres; *división y fusión de regiones*, que corresponde a un método genérico de segmentación basado en regiones, en donde la imagen se segmenta en regiones disjuntas que también cumplen con cierto criterio de homogeneidad o similitud [6].

Compensación de movimiento

La reconstrucción para la compensación de movimiento necesita de la imagen en el tiempo t_0 y utilizando los parámetros de movimiento calcular la imagen estimada en el tiempo $t_0+\Delta t$, es decir

$$I'^{\,t_0+\Delta t}(x\ y)) = \text{Interpol}\{I^{t_0}(x+dx\ \ y+dy)\} \qquad (1.3)$$

en donde dy y dy están en función de los parámetros de los modelos de movimiento en las tablas 1.1 y 1.2, Interpol$\{f(x, y)\}$ es una interpolación bidimensional.

La técnica de reconstrucción se ilustra en la figura 1.2.

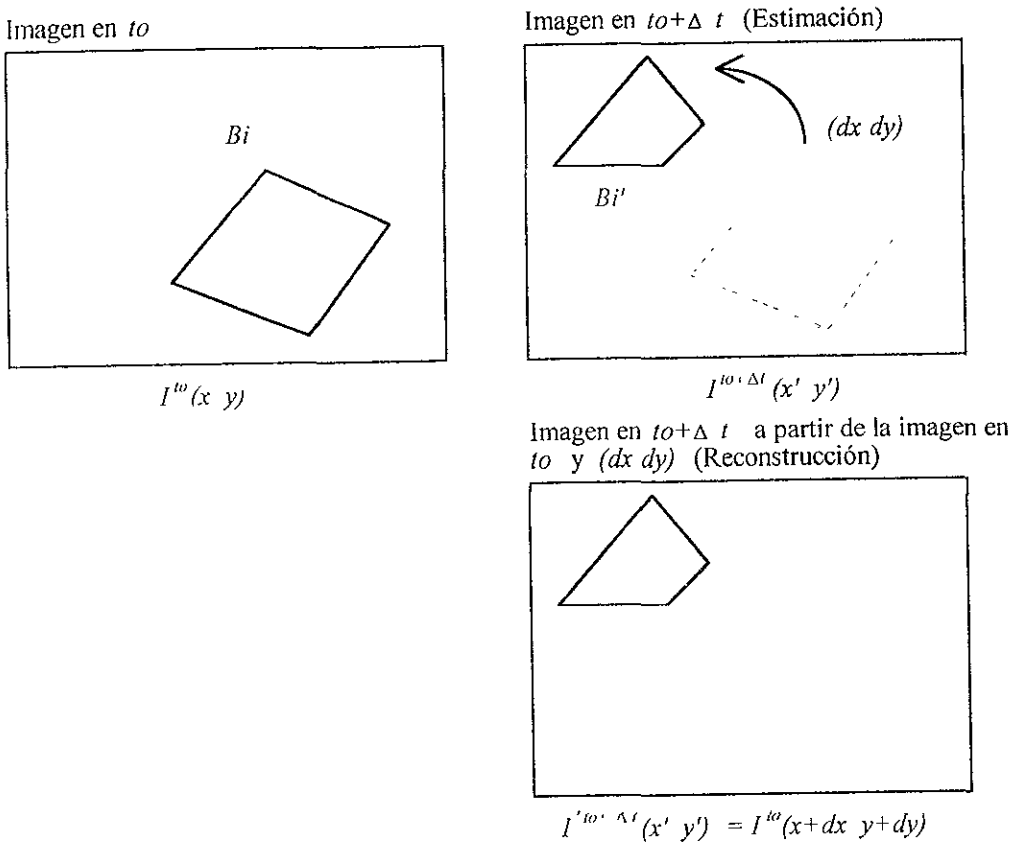


Figura 1.2. Compensación de movimiento.

Es decir, la reconstrucción consiste en la transformación plana del parche plano *Bi* al parche plano *Bi'*, introduciendo fenómenos como traslación, rotación, cambio de escala y deformación lineal, según el modelo de cámara u objeto elegido.

No obstante, el cálculo de la imagen reconstruida requiere una interpolación en dos dimensiones, que para el caso más común es bilineal [15] como se ilustra en la figura 1.3. Esta técnica supone que la iluminancia entre cuatro pixeles de una vecindad forman una función lineal que generalmente no tiene variaciones importantes o rupturas bruscas. Inevitablemente, la interpolación introduce desviaciones en la reconstrucción que se reflejan irremediablemente en la calidad del video comprimido.

En la figura 1.3, tenemos un enrejado de pixeles en donde

$$I(x, y) = (1-\Delta x)(1-\Delta y)I(i, j) + \Delta y(1-\Delta x)I(i, j+1) + \Delta x\Delta y I(i+1, j+1) + \\ + \Delta x(1-\Delta y)I(i+1, j) \qquad (1.4)$$

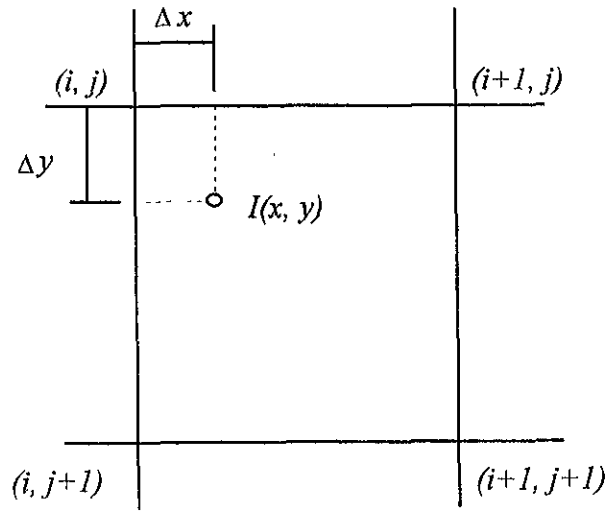


Figura 1.3. Interpolación bilineal.

también

$$i = \text{ENT}\{x\}$$
$$j = \text{ENT}\{y\}$$
$$\Delta x = \text{FRAC}\{x\}$$
$$\Delta y = \text{FRAC}\{y\}$$

La ecuación 1.4 puede verificarse fácilmente para los nodos del enrejado, al proporcionar valores particulares de i, j, Δx y Δy.

Codificación de movimiento

Para bajas tasas de bit, la eficiencia de la codificación en los parámetros de movimiento se convierte en un aspecto de particular importancia. Comúnmente existen dos tipos de descripciones acerca del movimiento: *parámetros de movimiento basados en regiones* y *vectores de movimiento basados en regiones*. El primero es adecuado para describir el campo de movimiento dentro de pequeñas regiones, mientras el segundo es más apropiado para movimientos grandes de regiones. Esto es debido a que el modelo de movimiento es válido para regiones pequeñas y es difícil encontrar un modelo que ajuste todos los vectores de una región grande.

Para especificar las descripciones acerca del movimiento basado en regiones existen dos tipos de información. La primera es la información de la región. Para la estimación de movimiento basado en regiones fijas, no se necesita información adicional a la región. Sin embargo, para la estimación de movimiento basado en regiones adaptables, tenemos que usar, por ejemplo, codificación de contornos para describir la región. Otros métodos eficientes son descriptores frecuenciales y métodos de aproximación poligonal.

El segundo tipo es como codificar eficientemente el movimiento para cada región. Para la descripción paramétrica del movimiento basado en regiones,

los parámetros de movimiento pueden codificarse al aplicar directamente codificación PCM (Pulse Code Modulation) o DPCM (Differential Pulse Code Modulation) entre regiones adyacentes. Para la descripción vectorial del movimiento basado en regiones, es imposible transmitir directamente los vectores de movimiento del campo denso. Para tal campo de movimiento, un método consiste en primero submuestrear el campo de vectores de movimiento y entonces utilizar DPCM para posteriormente comprimir los vectores restantes. Otro método consiste en aplicar técnicas directas para la codificación, como por ejemplo, codificación por árbol cuádruple, o código de cadena para comprimir los vectores del movimiento del campo denso. Es importante mencionar que la codificación de los parámetros de movimiento puede ser parte de la optimización de la estimación de movimiento.

1.2.2. Sistemas de compresión

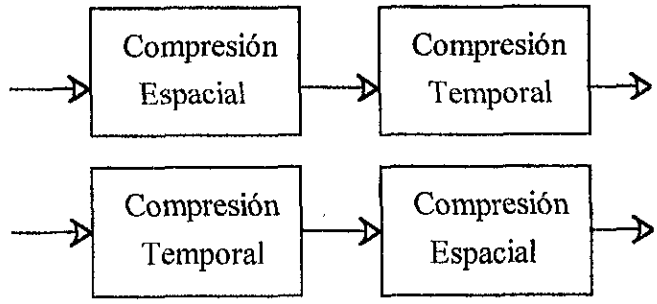
En un sistema de compresión de video a bajas tasas de bit se deben combinar técnicas de reducción en la redundancia espacial, temporal e inclusive espectral. Las diferentes combinaciones entonces especifican una gran variedad de sistemas de compresión. La elección de un sistema de compresión particular depende del ambiente de aplicación, por ejemplo si el ambiente es interactivo o unidireccional. Ambientes diferentes determinan diferentes requerimientos de codificación, tales como retardo de codificación y requerimientos de tiempo real de operación. En la siguiente discusión y en lo restante del trabajo nos enfocamos sólo a la compresión de la señal de iluminancia, lo que implica que la información de crominancia puede ser manejada por métodos convencionales que incluyen transformaciones espectrales y códigos fuente. Por lo tanto, la tarea principal es cómo combinar la compresión temporal y espacial para formar un sistema completo de compresión con alta eficiencia.

Existen dos tipos de combinaciones en la combinación de técnicas de compresión: *compresión espacial/temporal* y *compresión espacial-temporal*. La figura 1.4 muestra las técnicas. La primera es de un estilo híbrido, también conocida como *codificación híbrida*, mientras que la segunda es de un estilo conjunto conocida como *codificación 3D*.

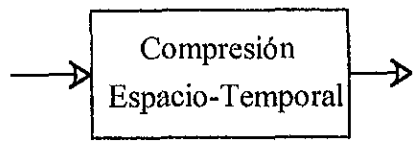
Compresión espacial y temporal híbrida

En la compresión híbrida se combinan dos técnicas diferentes de manera que las características atractivas de ambas pueden utilizarse. Las combinaciones de los dos bloques de codificación en ordenes diferentes especifican dos esquemas híbridos. El primer tipo, al que denotamos como *codificación híbrida II*, emplea primero un codificador por transformación seguido de un conjunto de codificadores DPCM actuando sobre los componentes codificados. El segundo tipo, denotado como *codificación híbrida I*, difiere del anterior en que el bloque de transformación es puesto en el interior del lazo de realimentación en el codificador predictivo. Los dos

tipos de esquemas híbridos son ilustrados en la figura 1.5. En ésta figura, T denota el bloque de transformación, Q denota el bloque de cuantización y T^{-1} y Q^{-1} denotan las operaciones inversas respectivamente. Además VLC (Variable Length Coding) denota la codificación de entropía, implementada normalmente como codificación de longitud variable.

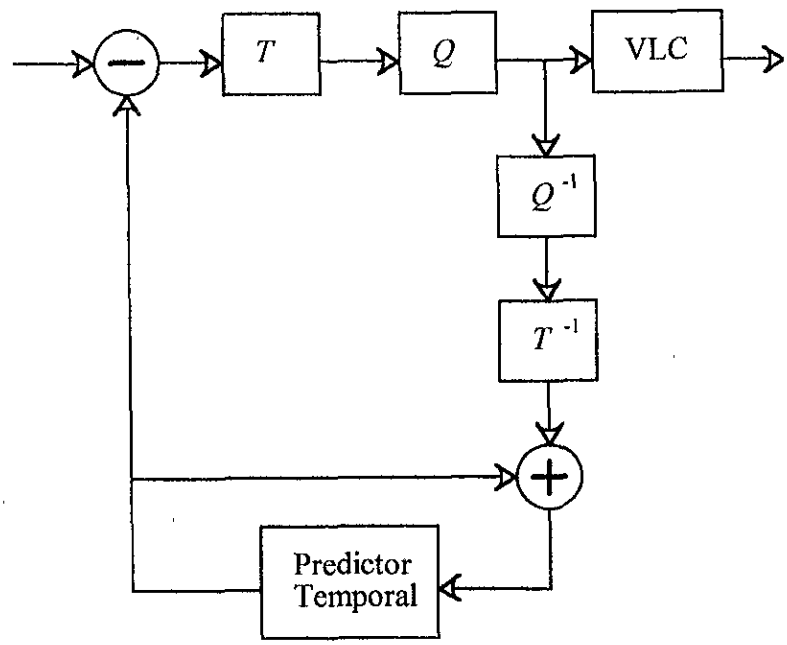


a)

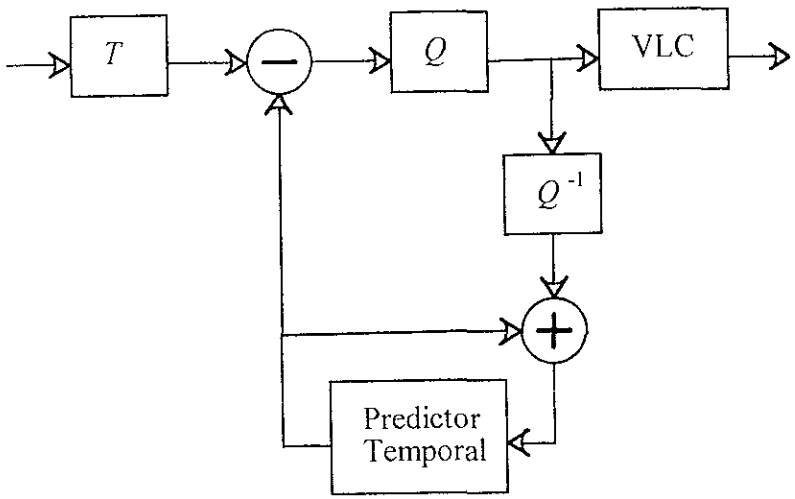


b)

Figura 1.4. Esquemas de compresión a) híbrida, b) 3D.



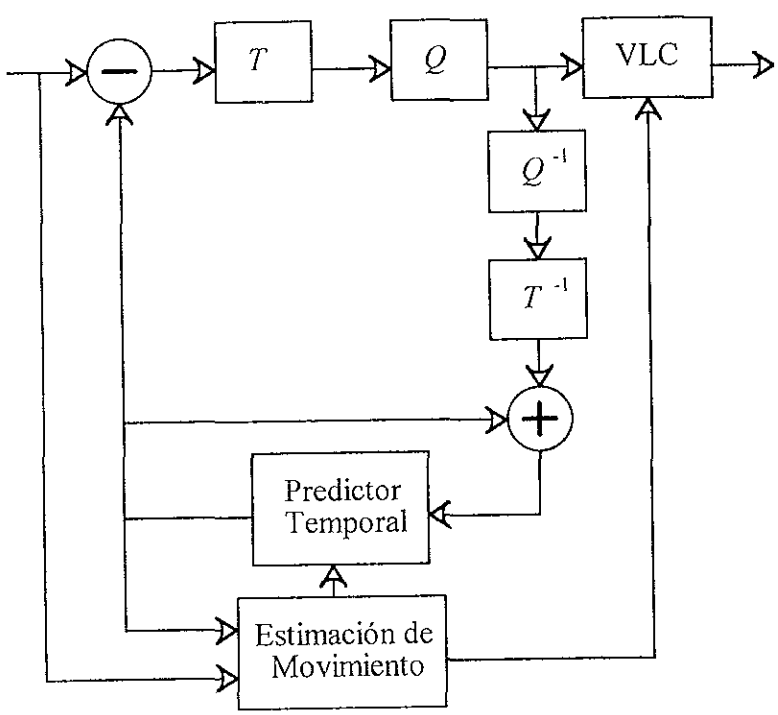
a)



b)

Figura 1.5. Compresión híbrida a) I, b) II.

a) Codificación híbrida I: Después de varios años de esfuerzos, la codificación híbrida I ha evolucionado hacia su versión actual, codificación híbrida III, con un esquema de compensación de movimiento DCT (Discrete Cosine Transform) como se muestra en la figura 1.6.



b)

Figura 1.6. Compresión híbrida III.

Este esquema ha gobernado actualmente los esfuerzos en los estándares de compresión abarcando desde emisiones terrestres de HDTV (High Definition Tele Vision), video digital (tal como en MPEGI), hasta el estándar para video teléfono H.261. Existen aún esfuerzos para usar tal esquema en la transmisión de escenas de video teléfono sobre la red telefónica actual. Las principales razones atrás de la popularidad de éste esquema residen en:

•Alta eficiencia de codificación. La redundancia temporal es explotada por el predictor de movimiento y la correlación espacial existente en la diferencia de movimiento compensado es removida usando DCT.

•Técnicas maduras de codificación. Es bien conocido que las técnicas DCT y predicción de movimiento son perfectamente maduras.

•Retardo de codificación corto. En muchas situaciones, el retardo de codificación puede estar estrictamente limitado. En relación con el punto anterior, las implementaciones hardware permiten retardos casi nulos.

Sin embargo, la codificación híbrida I no puede aplicarse directamente, en bajas tasas de bit, pero puede ser usada con algunas modificaciones. Algunas razones que impiden bajas tasas de bit, son las siguientes:

•Las estrategias de asignación de bits son las mismas para la imagen completa. Para bajas tasas de bit, se debe imponer cuantización dura en el sentido de evitar sobreflujo de almacenamiento temporal. Esto conduce a calidades de imagen pobres.

•Bajas tasas de bit permiten únicamente actualizaciones dispersas. Debido a la naturaleza de realimentación, los errores de reconstrucción se acumulan en cuadros consecutivos. Esto resulta en distorsiones graves.

•Las predicciones para el movimiento compensado generadas a partir del modelo basado en regiones, sufren problemas locales con patrones correspondientes a las regiones. Cuando la señal de error no puede transmitirse, o sólo pocas correcciones pueden transmitirse para permitir bajas tasas de bit, las imágenes reconstruidas muestran serias degradaciones.

Estas son las desventajas que producen calidades de imágenes pobres cuando la técnica se intenta utilizar en bajas tasas de bit. En el sentido de vencer los problemas mencionados, se sugiere que el sistema sea modificado mediante lo siguiente:

•Emplear técnicas de estimación y compensación de movimiento avanzadas. La estimación avanzada proporcionará campos de movimiento mejorados al realizar procesamientos de áreas ocultas, nuevas regiones, seguimiento de regiones y procesamiento de contornos ajustables. Adicionalmente, los alineamientos procesados por la compensación de movimiento pueden proporcionar mejor calidad de imagen al no considerar solapamientos o descubrimientos de regiones excesivos.

•Utilizar algoritmos de codificación espacial eficientes. Existen dos alternativas principales al utilizar la imagen de error. Una es comprimir

directamente la imagen de error y la otra es utilizar la imagen de error para direccionar áreas que necesitan ser transmitidas.

•Jerarquizar el peso de la codificación. Existen medidas para la cantidad de información que deben ser aplicadas a los *parámetros* de la compresión, de manera que se proporcione más peso a los parámetros que influyen directamente en la calidad de la imagen.

b) Codificación híbrida II: Existe un problema al incorporar compensación de movimiento en el dominio de la frecuencia ya que su dominio natural es el espacio Existen dos alternativas para tomar ventaja de la información de movimiento. La vía de la fuerza bruta para proporcionar compensación de movimiento se muestra en la figura 1.7 como esquema híbrido IV.

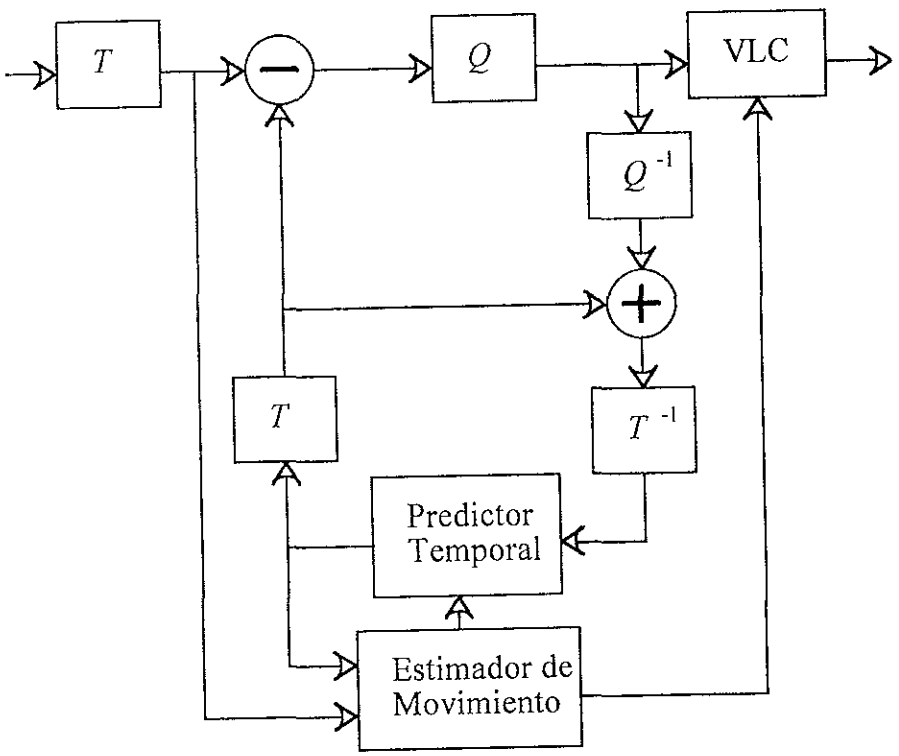


Figura 1.7. Compresión híbrida IV.

En comparación con el esquema III, la complejidad del esquema IV ha sido aumentada ya que existen dos bloques DCT y uno DCT inverso. La segunda ruta es desarrollar un algoritmo de estimación de movimiento que pueda trabajar en el dominio espectral. Dejando a un lado la complejidad en los circuitos, la principal ventaja del esquema es que operaciones diferentes como estimación de movimiento y compensación de movimiento pueden optimizarse conjuntamente debido a que se calculan en el mismo dominio.

Compresión espacial-temporal

Un enfoque natural es aplicar codificación espacio-temporal 3D directamente al espacio (x, y, t). Por lo tanto las redundancias espaciales

verticales y horizontales y temporales en el dominio espacio-temporal pueden explotarse en conjunto. El primer esquema de codificación 3D fue propuesto en 1977. Después de un corto periodo prospero, gradualmente fue ignorado debido a la popularidad de los esquemas híbridos, pero recientemente ha recibido atención creciente. Varios esquemas 3D (por ejemplo codificación sub-banda, codificación fractal, codificación por ondeletas) se han investigado para altas y medias tasas de bit en transmisión de video. A diferencia de la codificación híbrida en donde el movimiento de los entre-cuadros es tratado separadamente por las técnicas de compensación de movimiento, la información de movimiento en codificación 3D no es explícita. En lugar de ello, el movimiento es manipulado al sumar la coordenada de tiempo t en la transformación. Sin embargo, la codificación 3D sólo es efectiva en el caso de imágenes con pequeños movimientos. Cuando se presentan grandes movimientos entre cuadros sucesivos, la correlación temporal decrece fuertemente. Esto afecta dramáticamente la eficiencia del sistema. Por lo tanto, en el sentido de incrementar la eficiencia, debemos emplear información de movimiento para incrementar la correlación entre cuadros. Una idea es incorporar compensación de movimiento con el codificador 3D.

Sin embargo, no es sencillo combinar la compensación de movimiento con el codificador 3D, ya que la compensación requerida aquí debe ser invertible. Bajo la suposición de campo de movimiento uniforme, se puede proponer un esquema aproximadamente invertible de compensación basado en un submuestreo del campo de movimiento.

Aunque ambos esquemas de codificación híbrido y 3D son herramientas eficientes para la compresión de imágenes en movimiento, estos presentan diferentes comportamientos en ambientes de transmisión sin perdidas (por ejemplo emisiones de video). La característica principal de tales ambientes es la imposibilidad de garantizar que todos los receptores tendrán valores de predicción idénticos a los del transmisor. Sin embargo, la estructura del esquema híbrido demanda que el receptor permanezca alineado con el transmisor. Por lo tanto, un esquema híbrido no es normalmente aplicable en su forma pura. En contraste, la codificación 3D es ausente de error de realimentación debido a la estructura no recursiva del decodificador. Esto hace a la codificación 3D una técnica robusta en ambientes sin pérdidas. En el caso de codificación para el almacenamiento de medios, el esquema de codificación 3D facilita el acceso aleatorio de los datos. Características adicionales tales como búsqueda rápida o reproducción en reversa se pueden proporcionar sin costo extra.

1.3. Compresión basada en modelado

Si podemos reconstruir el modelo tridimensional de la escena que conduce a la secuencia 2D de imágenes, y las imágenes son analizadas y sintetizadas basadas en tal modelo, entonces se puede esperar un reducción importante de redundancia espacial y temporal. Esta es la idea básica atrás del codificador basado en modelado de la figura 1.8.

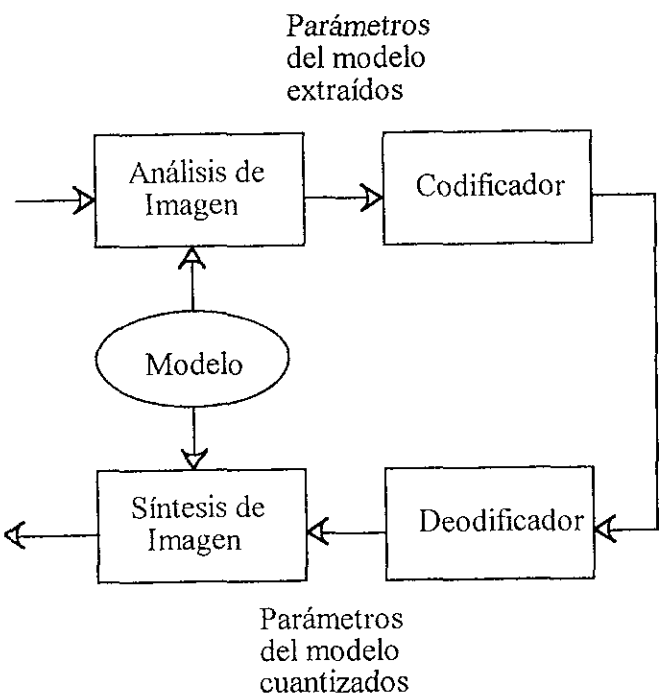


Figura 1.8. Diagrama a bloques de la codificación de video basada en modelado.

Los tres elementos clave son el *modelo*, que es la reconstrucción 3D de la escena, el *análisis de imagen*, mediante el cual la escena se adapta a los parámetros del modelo y la síntesis de imagen, proceso que invierte los parámetros del modelo hacia la secuencia de imágenes original.

El modelado normalmente consiste de dos partes: *modelo estructural* y *modelo de movimiento*. De acuerdo a la representación de la información estructural y de movimiento, los modelos pueden dividir a grosso modo en dos grupos: modelos *explícitos* e *implícitos*. En el primero, la estructura 3D y movimiento de la escena son explícitamente modelados mientras que en el segundo éstos sólo son tomados en cuenta implícitamente, por ejemplo al implicarse en el modelo 2D. Empleando un modelo 3D explícito las imágenes 2D pueden ser descritas con mayor exactitud y manipuladas con mayor facilidad. Adicionalmente, con modelos 3D las zonas ocultas pueden removerse de la imágenes proyectadas 2D, lo que es muy difícil conseguir

utilizando sólo modelos 2D. Sin embargo, en algunas situaciones, por ejemplo cuando la estructura explícita de las escenas puede no ser accesible o es difícil de describir, entonces el modelo implícito puede ser más favorable. Al utilizar modelo implícito, no se consideran completamente la estructura y el movimiento de los objetos sino su proyección en el plano imagen. Esto significa que sólo una descripción que refleja la conexión entre dos imágenes sucesivas en el plano imagen puede obtenerse para cada objeto. En tal descripción, ambas informaciones de estructura y movimiento son mezcladas en alguna forma paramétrica. Ya que el empleo de modelos implícitos es equivalente al uso de modelos avanzados descritos en la sección anterior, en lo siguiente sólo se consideran modelos 3D explícitos.

1.3.1. Modelado de imagen

En ésta sección se describen los modelos que son potencialmente aplicables y sus técnicas de modelado relacionadas.

Modelos de imagen

Ya que la descripción de movimiento está relacionada a la estructura de descripción utilizada, analizamos principalmente modelos geométricos para la descripción estructural. Los modelos geométricos pueden clasificarse en *descriptores basados en volumen* y *descriptores basados en superficies*, de acuerdo a las primitivas base, o en *descriptores paramétricos* y *descriptores no paramétricos*. La tabla 1.3 se refiere a la clasificación mencionada.

Ejemplos	Modelo basado en superficie	Modelo basado en volumen
Modelo paramétrico	Spline Superficie armónica	Cilindro generalizado Superquadrics
Modelo no paramétrico	Esqueleto de alambres	Voxel

Tabla 1.3. Clasificación de los modelos de escena.

La ventaja mayor de la descripción de escena basada en primitivas de superficie es que son fácilmente convertidas en representación superficial que favorece el sombreado (rendering). En la descripción basada en superficie, la descripción más popular es el *modelo no paramétrico mediante esqueleto de alambres*. En éstos modelos, la superficie es aproximada por parches planos poligonales (por ejemplo, parches triangulares). En el modelo de parches triangulares, la forma de la superficie es representada por un conjunto de puntos definiendo los vértices de los triángulos. Ya que el tamaño de los parches es ajustable de acuerdo a la complejidad de la superficie, la representación del esqueleto de alambres es flexible, general y, por lo tanto, ampliamente utilizada. La figura 1.9 muestra un esqueleto de alambres del cuerpo humano.

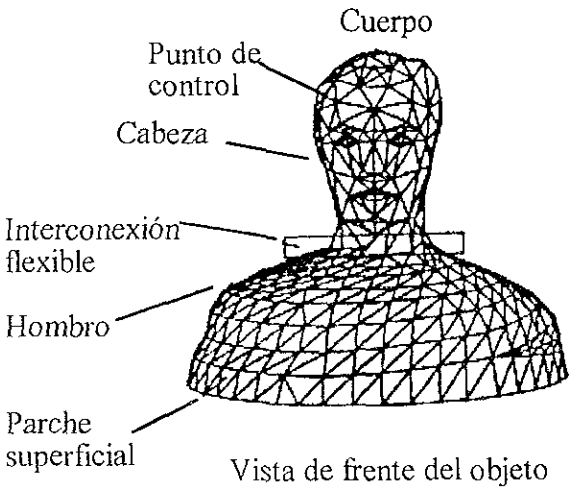


Figura 1.9. Modelo en esqueleto de alambres para el cuerpo humano.

Es importante hacer notar que la flexibilidad y generalidad del modelo puede convertirse en desventaja, si un objeto implica ocultamiento, estructura implícita o restricciones de movimiento. Un buen enfoque para vencer la desventaja es utilizar restricciones que especifiquen la flexibilidad de grupos de vértices. Por ejemplo, *unidades de acción* (AU) son utilizadas para describir la deformación en los musculos faciales. Para cierto tipo de objetos que tienen formas geométricas regulares, pueden ser más apropiados descriptores de superficies paramétricas, por ejemplo spline o superficies armónicas.

Muchos objetos del mundo real son sólidos, en cambio usualmente sólo sus superficies son visibles. Por lo tanto, una descripción basada en primitivas volumétricas es un enfoque natural al modelar objetos. Muchas investigaciones en modelos basados en volumen se centran en la descripción paramétrica del volumen. Un conjunto de primitivas frecuentemente usado es la clase de *cilindros generalizados* (GC). GC fueron usados en trabajos biológicos recientes en el análisis de ajuste de datos reales produciendo una descripción. Un ejemplo es inferir la adhesión de esqueletos de figuras a formas biológicas para su uso en apareamiento de modelos. Recientemente, Terzopoulos, Witkin y Kass han mejorado los GC con parámetros de deformación para el control de la elasticidad en el eje principal y en las paredes del cilindro. Utilizando tal modelo de cilindros, ellos son capaces de generar modelos 3D de algunos objetos naturales, tales como una papa, a partir de figuras 2D. Otra alternativa a los GC son los *supercuadrics*. Los supercuadrics y sus variaciones con deformaciones globales y locales contemplan una amplia variedad de formas naturales. Este tipo de estructura es capaz de modelar movimiento no rígido ya que cualquier deformación no rígida puede lograrse al empujar, pellizcar y jalar en un bulto de material clástico como arcilla. Pocos intentos se han realizado para construir modelos *no paramétricos basados en volumen*, debido a la popularidad de las gráficas de superficies. En éste modelo el conjunto de datos del volumen es

representado como un enrejado regular 3D discreto de voxels y normalmente es almacenado en una pila. Una comparación entre representaciones basadas en superficies y basadas en volumen se muestra en la tabla 1.4. Es notorio que los modelos no paramétricos basados en volumen pueden ser más apropiados para la codificación basada en modelado que los modelos basados en superficies.

Capacidad	Gráficas de superficie	Gráficas de volumen
Desempeño de sombreado	Sensible a la escena y a la complejidad de los objetos	Insensible a la escena y a la complejidad de los objetos
Requerimientos de memoria y procesamiento	Variable: depende de la escena y complejidad	Grande pero constante
Solapamiento de espacio-objeto	Ninguno	Frecuente
Transformación	Continuo: Desarrollado en la definición geométrica de los objetos	Discreta: desarrollada en sub-volumen
Conversión de rastreo y sombreado	Pixelización es incluida en la vista	Voxelización es independiente de la vista
Operaciones boleanas y de bloque	Difícil: debe desarrollarse analíticamente	Trivial: al usar voxel por voxel, árbol octal
Sombreado de interiores y fenómenos amorfos	No: sólo superficies	Si: sombreado de superficies y estructuras internas
Adecuación de datos muestreados e intercambio con datos geométricos	Parcial e indirecta (ajuste seguido de sombreado de superficies)	Soporta representación y sombreado directo
Mediciones (distancia volumen, normal)	Analítica: puede ser compleja	Aproximación discreta pero simple
Dependencia del punto de vista	Requiere calculo para cada cambio en el punto de vista	Pre-calcula y almacena información independiente del punto de vista

Tabla 1.4. Comparación entre gráficas de superficie y volumen.

Técnicas de modelado

Los modelos tridimensionales normalmente son generados a partir de mediciones o de conocimiento previo. Las mediciones más comúnmente empleadas son datos de rango o 2D, tales como intensidades de imagen o contornos. El tema de ésta sección es el modelado basado en mediciones disponibles.

a) Modelado estático: Existen dos enfoques para lograr modelado estático: modelado directo o iterativo. En el primero, los modelos son directamente construidos a partir de los datos disponibles. Específicamente, cuando los modelos paramétricos, ya sea basados en descriptores de volumen o superficie, son empleados el modelado es un problema de estimación paramétrica que puede ser transformado a un simple problema de minimización numérica.

Ya que los parámetros del modelo no son medibles en la práctica, no existe una forma directa para verificar el desempeño de los parámetros estimados. Un método alternativo es proporcionar una ruta de realimentación para el análisis de la imagen al emplear síntesis de imagen en el lazo de análisis. De ésta manera, la diferencia entre la imagen de entrada y la sintetizada es utilizada para refinar el ajuste al modelo de la escena. Este proceso se repite iterativamente hasta que se obtiene un resultado satisfactorio. La estrategia se muestra en la figura 1.10.

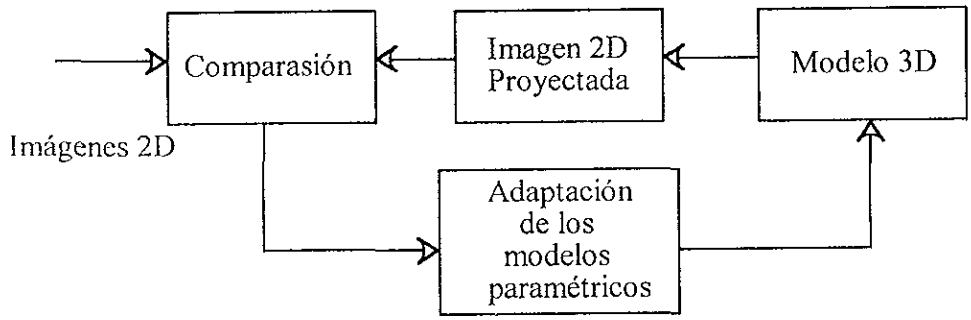


Figura 1.10. Modelo iterativo.

b) Modelado dinámico: No obstante, la construcción de un modelo 3D mediante análisis estático se vuelve inexacto o ineficiente ya que las mediciones pueden estar contaminadas o sólo partes del objeto están expuestas en algunos cuadros de la imagen. En tal situación es deseable fusionar las mediciones disponibles de la secuencia de imágenes a un modelo consistente. El problema clave yace en como lograr la fusión de información. Un enfoque para refinar el modelo de esqueleto de alambres es utilizar una herramienta estándar, filtrado de Kalman, al combinar mediciones obtenidas en diferentes tiempos. El proceso de fusión de la información puede ser aplicado sólo a una superficie existente. Cuando aparecen nuevos objetos, el modelo del objeto debe ser extendido basándose en nuevas mediciones.

1.3.2. La utilidad de los modelos

En muchas aplicaciones, es de interés construir modelos para formas naturales, por ejemplo formas biológicas u objetos hechos por el hombre. Formas biológicas como el cuerpo humano son descritas naturalmente por combinaciones boleanas jerárquicas de primitivas básicas. Una representación jerárquica para formas de animales fue sugerida por Marr y Nishihara (Figura 1.11).

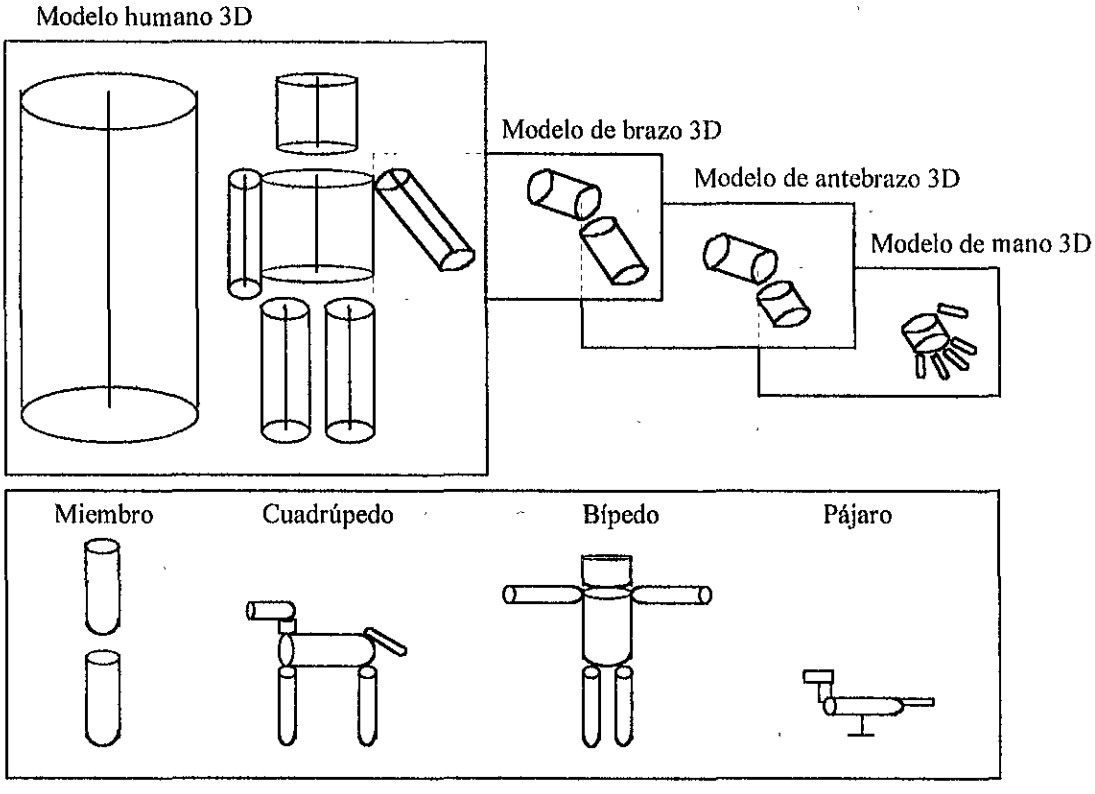


Figura 1.11. Descripción mediante GC.

Posteriormente, Pentland desarrollo una representación natural de formas dentro de una *representación con partes*. Este es un sistema de representación que proporciona un vocabulario de modelos y operaciones que permite modelar el mundo como una composición relativamente simple de *partes*. Él sugirió el uso de supercuadrics para describir las partes. Un ejemplo de una escena natural se muestra en la figura 1.12.

Una razonable y exacta composición de una cabeza humana compuesta por tan sólo 13 primitivas se muestra en la figura 1.13. Sólo 100 bytes de información fueron requeridos para especificar las primitivas.

Una alternativa es utilizar esqueletos de alambre para modelar primitivas o partes en lugar de modelos paramétricos. Por ésta razón, los esqueletos de alambres son herramientas principales para modelar formas biológicas complejas. La mayor desventaja es que el análisis de imágenes (por ejemplo

remoción de superficies ocultas) se torna complejo. Es claro que es deseable usar esqueleto de alambres para la síntesis de imágenes pero no para el análisis. En contraste, los modelos paramétricos son más apropiados para el análisis.

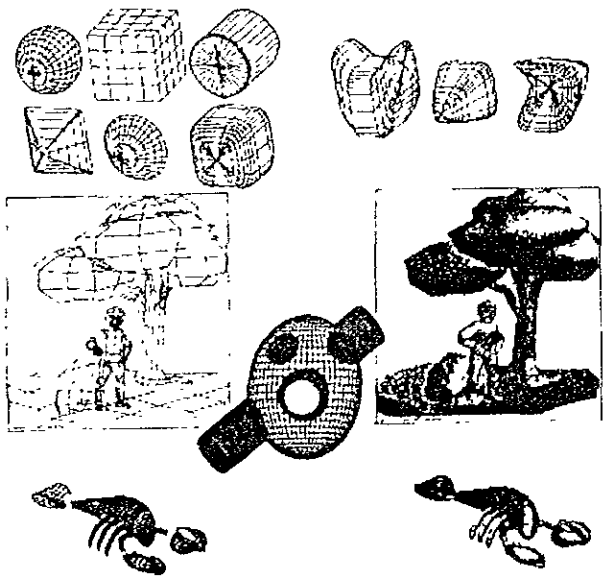


Figura 1.12. Escena natural combinando supercuadrics.

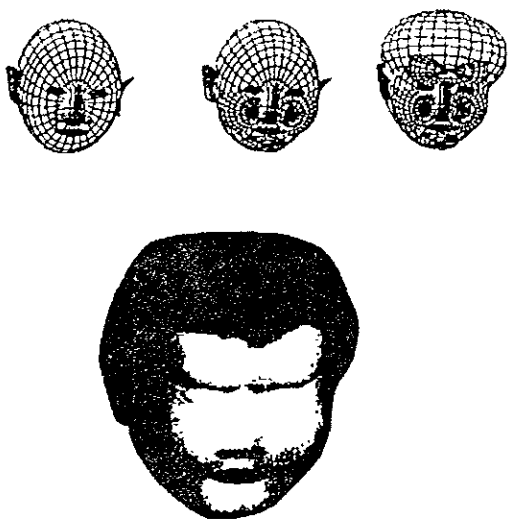


Figura 1.13. Cabeza humana artificialmente sintetizada usando la combinación de supercuadrics.

Como fue mencionado, los GC no pueden describir sucinta y exactamente las formas naturales o producir en forma sucinta formas inanimadas complejas. No obstante su falta de naturalidad a sido mejorada mediante el uso de supercuadrics en la síntesis de imágenes naturales, en cambio las imágenes reconstruidas carecen de realidad aunque se usen mapas de textura. Por lo tanto los modelos paramétricos son sólo capaces de descripciones extremadamente abstractas de las más naturales formas biológicas. Los

modelos paramétricos son ampliamente utilizados en tareas de análisis de imágenes, tales como rastreo, o en el modelado de objetos inanimados construidos por el hombre, tales como edificios. El problema de combinar modelos paramétricos con no paramétricos en el sentido de ampliar el realismo de las escenas sigue siendo un área de trabajo importante.

1.3.3. Codificación orientada al modelado

La idea básica atrás de la codificación basada en modelado es reconstruir un modelo global que tenga un modelo de imagen idéntico a la imagen original. El modelo global consiste de objetos, en donde cada objeto está especificado por tres conjuntos de parámetros definiendo información de movimiento *Ai*, forma *Mi*, y color *Si*. El diagrama a bloques de la codificación basada en modelado se muestra en la figura 1.14. Los tres bloques funcionales importantes son: *modelos de objeto*, *análisis y síntesis de imagen* y *codificación de parámetros*.

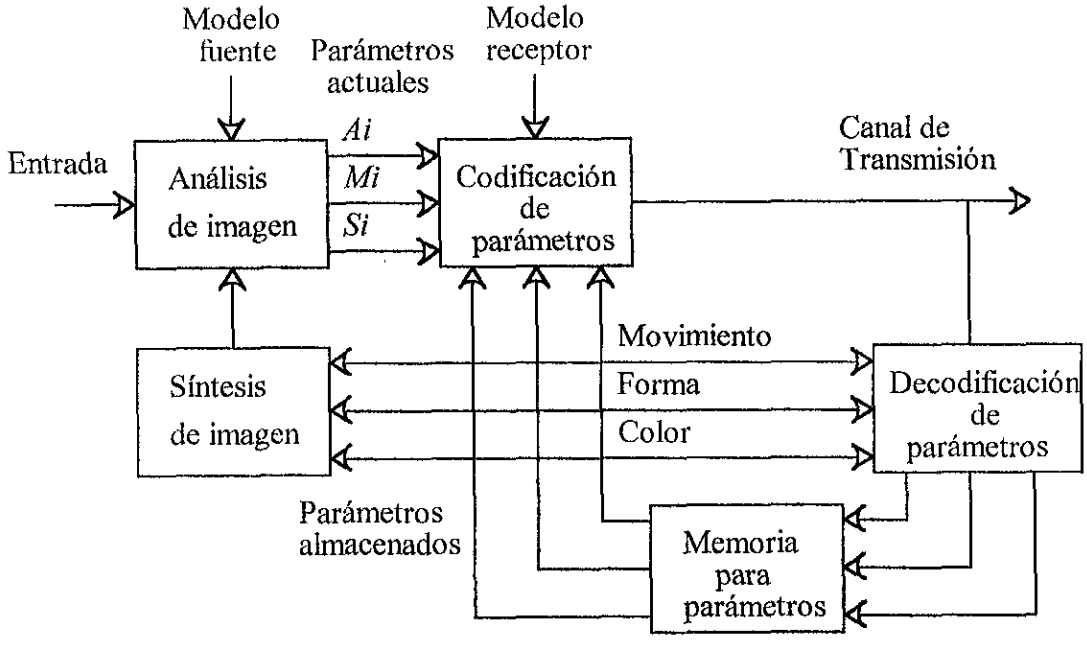


Figura 1.14. Diagrama a bloques de la codificación orientada a modelado.

Modelos de objeto

La eficiencia de la codificación orientada a modelado depende fuertemente de los modelos de objeto asumidos. Cuatro modelos de objeto son de interés especial: *objeto rígido 2D con movimiento 3D*, *objeto flexible 2D con movimiento 2D*, *objeto rígido 3D con movimiento 3D* y *objeto flexible 3D con movimiento 3D*. La evaluación revela que los modelos rígidos, o flexibles conectados con modelos de componentes rígidos no observan eficiencias de codificación altas. Se ha comprobado que al asumir objetos rígidos no se conserva la naturalidad de la escena. En contraste, casi todos los cambios de

imágenes sucesivos debidos a el movimiento o deformaciones en los objetos, son modelados cercanamente a la perfección mediante modelos flexibles 2D. Cuando un modelo de movimiento flexible 2D es utilizado en lugar de un modelo de movimiento rígido 3D, el área de falla en el modelo se reduce del 9% al 4% con respecto al área de la imagen. Similarmente, basado en la suposición de objetos 3D moviéndose arbitrariamente en el espacio 3D, se propone un modelo flexible que consiste de elementos rígidos 3D que están conectados flexiblemente. Al utilizar los nuevos modelos flexibles 3D, el área promedio de falla se puede reducir de un 4% a un 3% con respecto al área total de la imagen original. La eficiencia de codificación en modelos de objeto diferentes se ilustra en la tabla 1.5.

Modelo fuente	Información de movimiento	Información de forma	Información de color
Objetos 2D rígidos, Movimiento 3D	600	1300	1500 rs
Objetos 2D flexibles, Movimiento 2D	1100	900	4000 rs
Objetos 3D rígidos, Movimiento 3D	200	1640	4000 rs
Objetos 3D flexibles, Movimiento 3D	650	1550	2800 rs

Tabla 1.5. Tasas de bit promedio para varios modelos fuente, rs: tasa de bit asignada por pel.

Análisis y síntesis de imagen

La tarea en el bloque de análisis de imagen es estimar los tres conjuntos de parámetros, movimiento *Ai*, forma *Mi*, y color *Si* para cada objeto *y*. La complejidad en el análisis de imagen depende del modelo de objeto adoptado. Cuando se emplea modelo 2D, no es necesario reconstruir la estructura 3D de los objetos, ya que el movimiento y la forma 3D están implícitos en el modelo 2D. Una gran ventaja del modelado implícito es la capacidad de combinar fácilmente segmentación y modelado de objetos. Si se adoptan modelos de objeto 3D, el análisis de imagen se torna más complicado, debido a que la estimación 3D del movimiento y forma de objetos a partir de imágenes 2D es un problema fuerte en visión por computadora. Una manera de eliminar la dificultad es emplear modelado de objetos para agregar análisis de imágenes, esto es, utilizar una estrategia de análisis basada en modelado. Para hacer trabajar a tal sistema de codificación, los modelos 3D de los objetos en la escena deben construirse. Normalmente, una primera aproximación de la escena se propone como una adivinanza. Mientras mejor sea la aproximación,

menor será el esfuerzo para corregir el modelo. A continuación se describe un método de dos pasos para la inicialización de modelos de objeto:

1. Extracción de siluetas 2D de los objetos,: Para tal tarea, se pueden utilizar detección en los cambios temporales, o, si se cuenta con secuencias de imágenes estéreo, la profundidad de campo estimada se puede utilizar para segmentar la forma 2D.
2. Inferir la forma 3D de las siluetas 2D: Un algoritmo puede derivar líneas de contorno de un objeto a profundidad constante a través de la erosión de la silueta del objeto.

Una vez que los modelos de objetos 3D se construyen, la operación de codificación se puede llevar a cabo. Ahora la tarea principal en el análisis de imagen es estimar el movimiento 3D y refinar los modelos iniciales. Simultáneamente a la estimación de movimiento y forma existe el problema de resolver un número grande de sistemas de ecuaciones. Entonces se plantea una técnica de simplificación que trata separadamente las estimaciones en forma iterativa.

Una vez que los conjuntos de parámetros están disponibles, las imágenes 2D pueden ser reconstruidas por el bloque de síntesis de imagen.

Comentarios

Las imágenes no pueden ser descritas enteramente por los modelos orientados a objetos. Por ejemplo, para las escenas de cabeza y hombro, aproximadamente del 91-97% de las áreas pueden modelarse. Por lo tanto, las áreas remanentes corresponden a partes perceptualmente importantes tales como ojos y boca. Esto implica que al emplear solamente métodos de codificación basados en objetos no sean suficientes. Las áreas de falla en el modelo deben manejarse con codificación basada en forma de onda. La razón del porqué la codificación orientada a objetos es empleada, está en una combinación exitosa con las técnicas basadas en forma de onda.

La codificación orientada a objetos trabaja bien a tasas de bit de 64 kb/s. Sin embargo, esto enfrenta un cambio serio cuando se utiliza para comprimir video a muy bajas tasas de bit, en el orden de 8-16 kb/s. De la tabla 1.5 se puede encontrar que cerca de 2 kb/cuadro por cuadro se deben gastar en áreas en consentimiento con el modelo y cerca de 4 kb/cuadro en áreas de falla en el modelo. Esto implica que, aún cuando se empleen modelos avanzados de codificación (por ejemplo modelos basados en predicción), es difícil conseguir tasas de bit de 2 kb/cuadro. Para resolver éste problema, una posibilidad es incorporar codificación basada en semántica (máscara facial) en una codificación orientada a objetos, como se ilustra en la tabla 1.6.

1.3.4. Codificación basada en semántica

La idea básica atrás de la codificación basada en semántica es extraer los parámetros del modelo en el transmisor y sintetizar la escena en el receptor

basándose en un modelo explícito 3D. La técnica puede utilizarse en aplicaciones restringidas como teleconferencia. La naturalidad de las escenas reconstruidas puede ser mejorada aplicando técnicas de sombreado y texturizado. Existe una gran variedad de técnicas similares a la codificación basada en semántica, en donde se busca profundizar en técnicas de estimación de movimiento, texturización, extracción de profundidad y combinación con modelos basados en forma de onda.

Modelo fuente	Información de movimiento	Información de forma	Información de color	Total Bits/cuadro
Objetos 3D flexibles, Movimiento 3D	200 (global) 450 (local)	550 MC 1000 MF	2800 rs	5300
Objetos 3D flexibles + Máscara facial	200 (global) 90 (local)	550 MC 300 MF	900 rs	2040

Tabla 1.6. Tasas de bit promedio para modelos fuente incorporados, rs: tasa de bit asignada por pel.

Como ya se mencionó, la codificación semántica se utiliza principalmente en aplicaciones restringidas, en donde sólo una clase limitada de objetos aparecen (por ejemplo en escenas de cabeza y hombros). Entonces el conocimiento previo y detallado acerca de los objetos puede ser explotado plenamente. Esto nos habilita a extraer la descripción semántica de los objetos. Un diagrama a bloques de tal esquema de codificación se presenta en la figura 1.15.

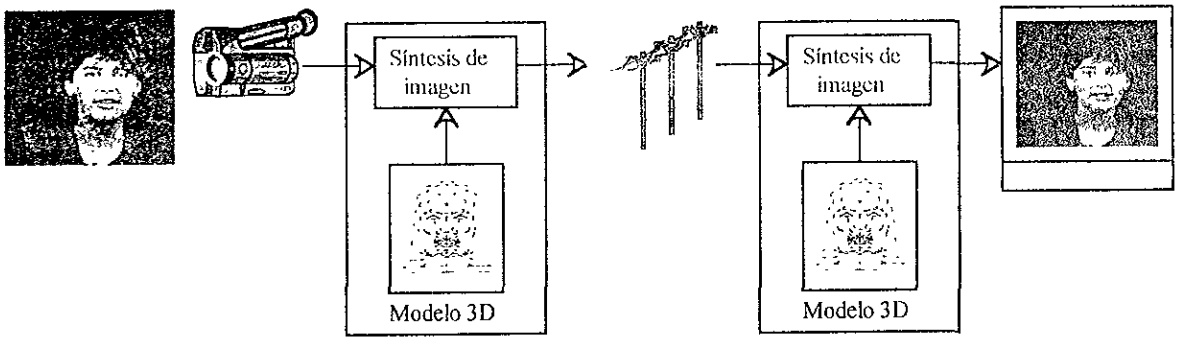


Figura 1.15. Codificación utilizando modelado semántico basado en objetos.

Se asume que ambos transmisor y receptor poseen el mismo modelo 3D facial y de textura al inicio de la sesión de comunicaciones. Durante la sesión se requiere que el lado transmisor extraiga los parámetros de movimiento facial y se mantenga actualizando los parámetros. Entonces en el lado receptor, la imagen es sintetizada usando los parámetros estimados de movimiento. Ya que sólo la información del análisis debe enviarse, éste tipo

de codificación puede realizar transmisión de video a bajas tasas de bit con alta calidad en la imagen. Tasas de bit típicas están alrededor de 0.5-1 kb/s.

Los aspectos claves en la codificación semántica para videotelefonía están en el análisis y síntesis facial.

Síntesis de imagen facial

La tarea de síntesis está en reconstruir una apariencia natural de la imagen facial utilizando un modelo facial. En el sentido de favorecer el sombreado a partir de un modelo 3D hacia una imagen 2D, los modelos faciales normalmente se basan en una representación en esqueleto de alambres. De acuerdo con las motivaciones atrás de las deformaciones en el esqueleto de alambres, los modelos faciales pueden agruparse en dos tipos:

a) Modelo facial geométrico: En éste tipo de modelo, la deformación en el esqueleto de alambres es puramente geométrica sin significado físico. La figura 1.16 muestra un ejemplo de los modelos faciales paramétricos comúnmente usados.

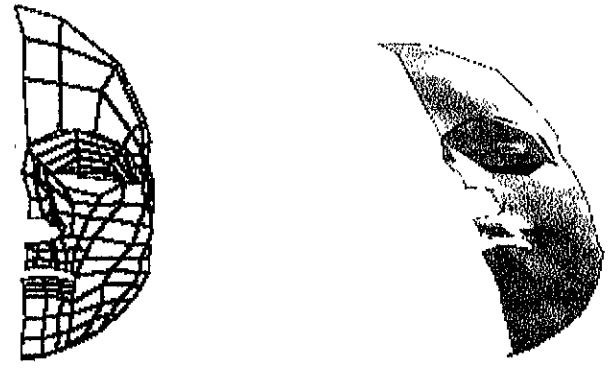


Figura 1.16. Modelo facial en esqueleto de alambres y sombreado.

Una ventaja importante de éste tipo de modelos es no solo su forma facial geométrica, también se proporciona una descripción de la expresión facial. Específicamente, las expresiones faciales son descritas por *unidades de acción* (AU), que están basadas principalmente en los *sistema de codificación faciales de acción* (FACS). Los AU esperan por un cambio pequeño en la expresión facial, y dependiendo de una pequeña activación muscular consiente, éstos lo expresan en forma paramétrica.

El movimiento puede modelarse mediante un modelo 3D geométrico facial. El movimiento facial 3D consiste de dos partes: movimiento global de la cabeza y variaciones locales de la expresión. Empleando AU, el movimiento puede ser descrito por

$$\mathbf{s'} = \mathbf{R}\,\mathbf{s} + \mathbf{T} + \mathbf{E}\,\Phi \qquad (1.5)$$

En donde $\mathbf{s} = (x, y, z)$ es un punto sobre la superficie, $\mathbf{R}$ es la matriz de rotación, $\mathbf{T}$ es el vector de traslación, $\Phi = (\phi_1, \phi_2, \ldots, \phi_m)^T$ son los parámetros

de movimiento facial local y **E** determina que tanto un punto **s** es afectado por Φ. Esto está determinado por las AU. En éste modelo de movimiento la suposición que se hace es acerca de la representación de las expresiones faciales como parte de una combinación lineal de AU.

b) Modelo facial físico: El modelo del rostro incorpora una aproximación al tejido facial humano que es modelado como una malla de puntos masa conectados por resortes con elasticidad no lineal. En el modelo de Terzopoulos y Waters, las variaciones en las expresiones son modeladas mediante contracciones musculares. Específicamente, el desplazamiento del nodo j desde $\mathbf{s}_j$ hacia $\mathbf{s'}_j$ debido a la contracción muscular es una suma ponderada de **m** actividades musculares actuando en el nodo j:

$$\mathbf{s'}_j = \mathbf{s}_j + \sum_{i=1}^{j} c_i \, b_{ij} \, \mathbf{m}_i \qquad (1.6)$$

en donde c_i *es un factor de contracción, y* b_{ij} es una función de mezcla muscular que especifica una zona radial de influencia para la fibra muscular.

Estos dos tipos de modelos faciales pueden utilizarse para controlar el movimiento en el rostro sintetizado. Para lograr una apariencia natural en la imagen facial, se deben aplicar técnicas de texturización al esqueleto de alambres.

Análisis de imagen facial

La tarea principal en el análisis de imagen facial es extraer parámetros importantes requeridos por la codificación, tales como posiciones de las características faciales, parámetros de movimiento, y profundidad. Algunos aspectos importantes son:

a) Localización de características faciales: La localización de características faciales es motivada por el menos tres aspectos. Primero, la localización es un paso necesario en el ajuste automático de un esqueleto de alambres a un rostro específico. Segundo, las características faciales son aspectos importantes en la estimación de movimiento. Y tercero, una importancia extra puede proporcionarse a las características faciales si estas características subjetivas pueden ser localizadas.

b) Estimación de movimiento facial: De acuerdo a si se emplea un modelo de rostro a priori, las técnicas de estimación de movimiento facial se pueden clasificar en dos tipos: *estimación de movimiento basado en un modelo facial* y *estimación de movimiento NO basado en un modelo facial*. En estimación de movimiento basado en modelado, una estrategia poderosa, análisis por síntesis, puede utilizarse para estimación de movimiento. Basándose en AU, el movimiento facial puede modelarse bien mediante (1.5). En éste sentido, la estimación de movimiento se reduce a un problema de identificación paramétrica. De acuerdo a que cantidad de medición se utiliza, se dividen los métodos de estimación paramétrica en dos tipos: *estimación paramétrica facial basada en flujo de imagen* y *estimación paramétrica facial basada en*

flujo óptico. Existen dos esquemas adicionales para manejar el movimiento global de la cabeza y las expresiones locales: El primer esquema es tratar separadamente los movimientos global y locales. Esto es, primero especificamos los parámetros locales a cero y estimamos los parámetros globales. Entonces se estiman los parámetros locales después de una compensación de movimiento global. El segundo esquema es estimar conjuntamente ambos movimientos global y local. Cuando se emplea modelo facial basado físicamente, necesitamos estimar dinámicamente las contracciones faciales musculares. Terzopoulos y Waters usaron modelos con contornos deformables (snakes) para seguir los movimientos no rígidos de las características faciales en imágenes de video. La interpretación automática en la deformación de los snakes conduce a parámetros dinámicos de musculo que permiten al modelo reconstruir movimientos faciales. En estimación de movimiento facial no basado en modelado, la estimación de movimiento opera directamente en imágenes 2D. Por ejemplo, el movimiento de musculos faciales puede estimarse en base a flujo óptico o extracción de puntos característicos al rostro y el rastreo o seguimiento puede realizarse al utilizar modelos con contornos activos. Sin la inclusión de la síntesis de imágenes en el bloque de análisis, el análisis de la imagen se vuelve simple.

Comentarios

a) Texturización facial: La texturización es un aspecto importante en codificación basada en semántica. La técnica de texturización consiste en sintetizar una imagen facial al deformar una textura original. Existen dos problemas al texturizar:

1. Ciertas expresiones faciales no pueden generarse, por ejemplo, el arrugamiento de la piel alrededor de los ojos y la boca.
2. No es suficiente utilizar una imagen sencilla aislada. Esto se debe a la imposibilidad de la textura de la imagen de tener simultáneamente la boca cerrada y abierta.

Una forma de salvar estos problemas es emplear ciertas texturas típicas de imágenes. Otra forma es emplear codificación basada en forma de onda al transmitir la imagen de diferencia entre la original y la sintetizada.

b) Integración de la información: La videotelefonía es un servicio visual que combina ambas informaciones acústica y visual. Los dos tipos de información están cercanamente relacionados. Un problema interesante es como emplear la información de voz para mejorar el análisis de expresión facial.

c) Información personal: Es creíble que el análisis de imagen relacionado a la codificación basada en semántica es difícil. Algunas veces, los problemas son innecesariamente complejos. Considere como ejemplo el ajuste del esqueleto de alambres al rostro del hablante. Esto es sin duda un problema difícil. Personas diferentes tienen diferentes formas faciales y tamaños. En el sentido de ajustar una persona especifica a un modelo genérico no tan sólo

debe ser cambiado en tamaño sino también en estructura. Sin embargo si se utiliza un esqueleto de alambres personal se reduce la dificultad en las tareas de análisis de imágenes.

Capitulo 2

Filtrado morfológico espacial

2.1. Generalidades

Gran parte del procesamiento digital de imágenes está enfocado al diseño de *filtros que desempeñan una tarea particular.* Algunos ejemplos de aplicaciones para el filtrado se encuentran en suavizado, detección de contornos, eliminación de ruido, y mejoramiento de la imagen. El análisis del comportamiento en el dominio del tiempo y la frecuencia es el enfoque adoptado en el diseño de filtros. No obstante, existen técnicas que son, en algún sentido, óptimas para un trabajo particular. Lo anterior se realiza al establecer un criterio de desempeño y entonces maximizar el criterio al elegir adecuadamente la respuesta al impulso del filtro.

La historia del procesamiento de imágenes ha sido en gran parte el diseño de *filtros. Los filtros son elegidos por razones de simplicidad* de cálculo, éxito comprobado, conveniencia, apariencia estética e inclusive recomendaciones y antojo. Tal enfoque de diseño puede probar su éxito, pero también se puede considerar subóptimo. El diseño subóptimo casi nunca produce la mejor tarea de filtrado y puede ser evidentemente peligroso.

Los filtros subóptimos -particularmente aquellos que son implementados fácilmente en la computadora- pueden introducir artefactos o remover componentes de interés en la imagen, usualmente sin precauciones. Los filtros que involucran el pulso rectangular bajo un dominio -filtros populares para los *programadores de computadoras- tienen un comportamiento no satisfactorio en* el dominio opuesto, debido a las ondulaciones infinitas de la función sinc(x).

Los usuarios de filtros de bordes cuadrados en un dominio están frecuentemente plagados de ruido y otros fenómenos indeseables en el otro dominio. Ellos algunas veces lamentan erróneamente las características indeseables como inherentes al procesamiento digital, o lamentan la falta de poder computacional necesaria para realizar el trabajo adecuadamente. Al considerar técnicas de diseño óptimo, en general, los filtros se comportan bien [5, 8].

Por definición, si un filtro falla la prueba de linealidad, éste es no lineal. Existen muchos tipos de filtros no lineales que pueden resolver mejor ciertos tipos de problemas en procesamiento de imágenes de los que pueden resolver los filtros lineales. Sin embargo, los filtros no lineales carecen de los grandes alcances y de la relativa facilidad en el marco teórico que subyace a los filtros lineales. Para un tratamiento introductorio del filtrado, existen los filtros llamados de *orden estadístico,* conocidos así debido a que están basados en estadística derivadas del ordenamiento de elementos de un conjunto en lugar de cálculos de medias, desviaciones estándar, etc. El filtro de mediana es un ejemplo.

De lo anterior y como es razonable suponer, el procesamiento digital de imágenes está basado tradicionalmente en los conceptos de sistemas lineales y análisis de Fourier (o de otras transformadas relacionadas). A pesar de que tales enfoques clásicos han sido fructíferos en muchas aplicaciones, están limitados en uso para *imágenes como señales,* ya que no enfrentan directamente el problema fundamental de cómo cuantificar la estructura geométrica o de forma en la señal. En contraste, la morfología matemática, que es una metodología teórica de conjuntos, puede rigurosamente cuantificar muchos aspectos de la estructura geométrica de las señales en una forma que concuerda que la intuición y percepción humana. Este método, que tiene sus orígenes matemáticos en teoría de conjuntos, geometría integral, análisis convexo, estereología, y probabilidades geométricas, fue desarrollado por Matheron y Serra [23] en los sesentas.

Las técnicas de la morfología matemática están basadas en conceptos de teoría de conjuntos, en superposiciones no lineales de señales, y en una clase de sistemas no lineales que llamamos *sistemas morfológicos.* Consideramos al término *morfología matemática* como una descripción más general, refiriéndose al cuerpo entero de teoría fundamental de sistemas morfológicos y a los algoritmos heurísticos asociados con la aplicación de la teoría en áreas especificas. La morfología matemática ha sido ampliamente utilizada para análisis de imágenes biomédicas y de microscopía electrónica, y ha sido una herramienta valiosa en muchas aplicaciones de visión por computadora, específicamente en el área de inspección visual automatizada. Las aplicaciones industriales de éstas técnicas se han impulsado por el continuo desarrollo y mejoramiento de arquitecturas de cómputo innovadoras que implementan transformaciones morfológicas de señales. Sin embargo, a pesar de su éxito en las aplicaciones y la profundidad y elegancia de su estructùra matemática, la morfología matemática sólo recientemente ha adquirido interés académico lamentando su escaso esparcimiento.

En el caso general, el procesamiento morfológico de las imágenes opera al pasar un *elemento estructurante* sobre la imagen, en una forma similar a la convolución (figura 2.1). Cómo la máscara de la convolución (kernel o núcleo), el elemento estructurante puede ser de cualquier tamaño, y puede contener cualquier componente de 1's o 0's. En cada posición de pixel, se desarrollo una operación lógica específica entre el elemento estructurante y la

imagen binaria o en escala de grises subyacente. Entonces, el resultado de la operación lógica se almacena en un arreglo para la imagen de salida. El efecto creado depende del tamaño y contenido del elemento estructurante y de la naturaleza de la operación lógica.

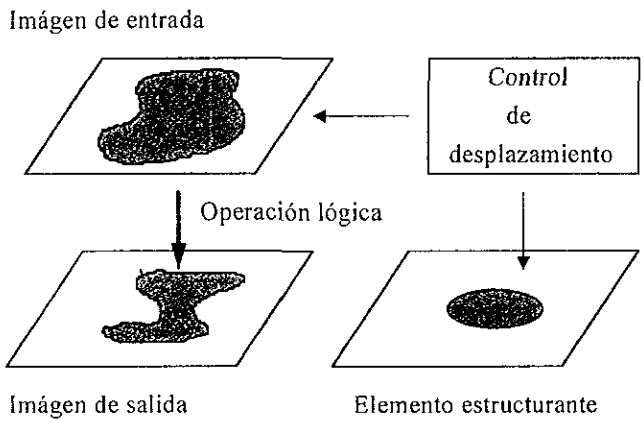


Figura 2.1. Procesamiento morfológico de imágenes.

Una inspección comprensiva del campo entero de la morfología matemática debe ser superficial aunque exista espacio disponible en el texto. Por lo tanto, nos concentramos en sistemas morfológicos, con los objetivos de mostrar cómo estos sistemas pueden enriquecer las herramientas para el procesamiento de imágenes e ilustrar como pueden ser aplicados a los campos de interés creciente tales como visión por computadora, filtrado no lineal, análisis estructural de señales, y en nuestro caso particular compresión de video.

Este capítulo examina la interpretación teórica y práctica de algunas transformaciones morfológicas que son usadas como herramientas de filtrado con objetivos concretos. En nuestro caso de compresión de video y como parte del preprocesamiento para la etapa de segmentación, el objetivo que planteamos es suavizar la imagen sin remover contornos. Es decir, eliminar detalles perceptualmente menos importantes sin remover las componentes de interés de alta frecuencia que definen bordes. Con ésta técnica buscamos disminuir la carga computacional en la etapa de segmentación, de manera que la segmentación no conduzca a resultados erróneos o imágenes sobresegmentadas.

El capítulo está organizado como se describe a continuación. La segunda sección revisa conceptos básicos acerca de los sistemas morfológicos. La tercera sección contiene aplicaciones al filtrado y al suavizado no lineal usando las transformaciones morfológicas *apertura y cierre*. La cuarta sección incorpora filtros mejorados para el suavizado sin eliminación de contornos, conocidos como *apertura y cierre por reconstrucción*. La quinta sección es un análisis comparativo entre los filtros presentados. Adicionalmente las secciones van acompañadas con ejemplos de implementación programados en lenguaje C.

2.2. Conceptos generales

La idea básica del enfoque morfológico al procesamiento multidimensional de señales es la representación de señales y sistemas en términos de un conjunto de transformaciones. Esto es la clave para representar y manipular estructuras geométricas en imágenes y otras señales.

2.2.1. Representación de señales

Dejemos que **R** y **Z** representen respectivamente el conjunto de los números reales y enteros, y tomemos a **E** como la *d*-dimensión (también representada como *d*D, en donde *d* es un entero) del espacio continuo $\mathbf{R}^d$ (d=1, 2, ...) o el espacio discreto $\mathbf{Z}^d$. Entonces una señal de *d*-dimensión puede representarse como una función cuyo dominio es cualquiera $\mathbf{R}^d$ (continuo) o $\mathbf{Z}^d$ (discreto), y cuyo rango es cualquiera **R** (amplitud continua) o **Z** (amplitud cuantizada).

Las señales binarias pueden representarse por conjuntos. Por ejemplo, la imagen de la figura 2.2.a es una señal binaria, en donde la región con fondo blanco puede representarse por 0 y la región sombreada puede representarse por 1. Es claro que la señal también puede representarse por el conjunto X de puntos correspondientes a la región sombreada. Las imágenes binarias se obtienen al umbralizar (comparar mediante un umbral) una señal en escala de grises. Al umbralizar se obtienen representaciones alternativas para las imágenes en escalas de grises en forma de conjuntos. Serra [23] utiliza la representación de una función real valuada en *d*-dimensión $f(\mathbf{p})$ ($\mathbf{p}$ es un vector *d*-dimensión) mediante el ensamble de sus conjuntos *d*-dimensión definidos por

$$T_a(f) = \{\mathbf{p}: f(\mathbf{p}) \geq a\} \quad -\infty<a<\infty \tag{2.1}$$

en donde la amplitud a se alterna toda por **R** o **Z** dependiendo si la señal f tiene rango continuo o cuantizado. Los conjuntos de umbrales tienen dos propiedades importantes: cercanamente ordenados ya que $a<b \Rightarrow Ta(f)\supseteq Tb(f)$, y pueden reconstruir únicamente la señal f ya que

$$f(\mathbf{p}) = \max\{a: \mathbf{p} \in T_a(f)\} \quad \forall \mathbf{p} \tag{2.2}$$

Esta representación se ilustra para una señal 1D por el ejemplo de la tabla 2.1. La señal $f(\mathbf{p})$, mostrada en el segundo renglón de la tabla, tiene sólo cuatro niveles de amplitud, y por lo tanto puede representarse por los cuatro conjuntos de umbral, cada uno de los cuales incluye sólo los puntos indicados por puntos en los siguientes cuatro renglones de la tabla (Una señal continua en amplitud requiere un número infinito de conjuntos de umbral). Los cuatro renglones finales muestran las señales binarias umbralizadas correspondientes a los conjuntos de umbral, en donde $f_a(\mathbf{p}) = 1$ si $f(\mathbf{p}) \geq$ a [por ejemplo , $\mathbf{p} \in T_a(f)$] y $f_a(\mathbf{p}) = 0$ si $f(\mathbf{p}) \geq$ a [por ejemplo , $\mathbf{p} \notin T_a(f)$], en donde a representa el

rango de $f(\mathbf{p})$. Obviamente, las señales $f_a(\mathbf{p})$ transportan la misma información que los conjuntos de umbral $T_a(f)$. Por lo tanto f puede ser reconstruida de f_a's ya que

$$f(\mathbf{p})=\max\{a: f_a(\mathbf{p})=1\} \quad \forall\, \mathbf{p} \tag{2.3}$$

La validez de las ecuaciones (2.2) y (2.3) se muestra en forma sencilla para el ejemplo de la tabla 2.1.

p	0	1	2	3	4	5	6	7	8	9	10
$f(\mathbf{p})$	1	1	2	1	3	0	0	1	0	2	3
$T_3(f)$					•						•
$T_2(f)$			•		•					•	•
$T_1(f)$	•	•	•	•	•			•		•	•
$T_0(f)$	•	•	•	•	•	•	•	•	•	•	•
$f_3(\mathbf{p})$	0	0	0	0	1	0	0	0	0	0	1
$f_2(\mathbf{p})$	0	0	1	0	1	0	0	0	0	1	1
$f_1(\mathbf{p})$	1	1	1	1	1	0	0	1	0	1	1
$f_0(\mathbf{p})$	1	1	1	1	1	1	1	1	1	1	1

Tabla 2.1. Representación por umbral de una señal cuantizada.

2.2.2 Transformación de señales

Las transformaciones de señales con morfología matemática, a las cuales llamamos filtros morfológicos, son operadores no lineales que modifican localmente las características geométricas de las señales multidimensionales. Primero consideraremos el caso de señales binarias. Tomemos $\mathbf{X}\subseteq\mathbf{E}$ como el conjunto de representación de la señal de entrada, y tomemos $\mathbf{B}\subseteq\mathbf{E}$ como un conjunto compacto de tamaño pequeño y forma simple (por ejemplo un circulo d-dimensión). El conjunto $\mathbf{B}$ es llamado elemento estructurante. Tomemos $\mathbf{X}\pm b = \{x\pm b: x\in\mathbf{X}\}$ como el vector de translación de $\mathbf{X}$ por $\pm b\in\mathbf{E}$. Los operadores morfológicos fundamentales para conjuntos son la *dilatación* $\oplus$ y la *erosión* $\otimes$ de $\mathbf{X}$ por $\mathbf{B}$, que están definidos como sigue

$$\mathbf{X}\oplus\mathbf{B} = \bigcup \mathbf{X}+b = \{x+b:\ x\in\mathbf{X} \text{ y } b\in\mathbf{B}\ \} \tag{2.4}$$

$$\mathbf{X}\otimes\mathbf{B} = \bigcap \mathbf{X}-b = \{z:\ (\mathbf{B}+z)\in\mathbf{X}\ \} \tag{2.5}$$

A partir de éstas definiciones, se puede mostrar que la salida del operador dilatación es un conjunto de puntos trasladados tales que la traslación del elemento estructurante reflectado $\mathbf{B}=\{-b:\ b\in\mathbf{B}\}$ tiene intersección no nula con el conjunto de entrada; es decir

$$\mathbf{X}\oplus\mathbf{B} = \{z:\ (\mathbf{B}+z)\ \cap\ \mathbf{X}\neq\varnothing\}$$

En forma similar, la salida del operador erosión es el conjunto de puntos trasladados tales que el elemento estructurante trasladado está contenido en el conjunto de salida.

Otros operadores pueden definirse como combinaciones de erosiones y dilataciones. Por ejemplo, dos operadores adicionales fundamentales son la *apertura* ○ y *cierre* ● de X por B, definidos como sigue

$$X \circ B = (X \otimes B) \oplus B \tag{2.6}$$

$$X \bullet B = (X \oplus B) \otimes B \tag{2.7}$$

Para visualizar el comportamiento geométrico de éstos operadores es conveniente considerar conjuntos 2D tales como el conjunto X y el elemento estructurante B de la figura 2.2. La figura 2.2 muestra que la erosión adelgaza al conjunto X, mientras que la dilatación expande a X. La apertura suprime las capas afiladas y corta las penínsulas angostas de X, mientras que el cierre llena los golfos estrechos y los hoyos pequeños, en forma tal que $X \circ B \subseteq X \subseteq X \bullet B$. Entonces, si el elemento estructurante B tiene una forma regular, ambos apertura y cierre pueden considerarse como filtros no lineales que suavizan los contornos de la señal de entrada. Es claro, que la forma y tamaño del elemento estructurante determinará la naturaleza y el grado de suavizado.

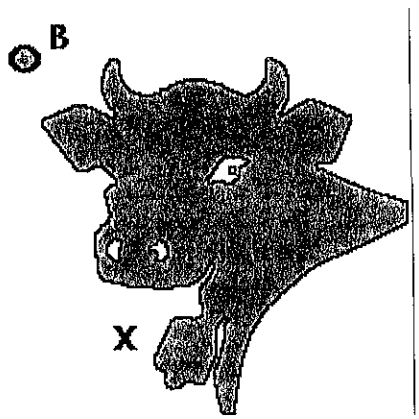
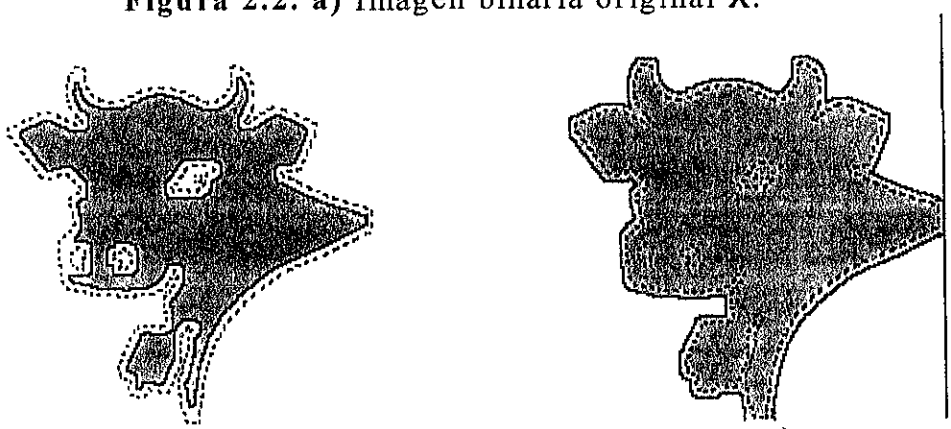
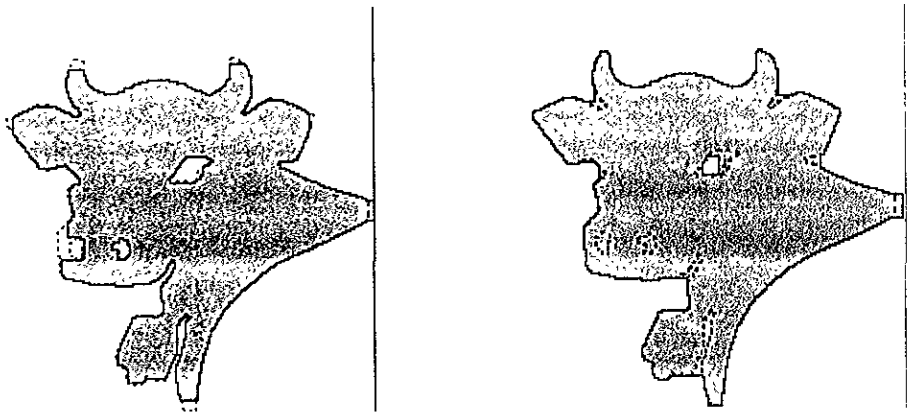


Figura 2.2. a) Imagen binaria original X.



b) **c)**

Figura 2.2. b) Erosión, **c)** dilatación de X por B.



d) e)

Figura 2.2. d) Apertura, **e)** cierre de X por B.

2.3. Apertura y cierre en escala de grises

Los operadores descritos arriba pueden extenderse a señales multinivel (por ejemplo, no binario), representadas por una función real valuada, de varias formas. Serra utiliza la representación de funciones $f(\mathbf{p})$ d-dim por una colección de sus conjuntos de umbral en la ecuación (2.1). Entonces, dilatar todos los conjuntos de umbral de f por el mismo conjunto compacto B proporciona los conjuntos $T_a(f)$ B, que son los conjuntos de umbral de una nueva función $f \oplus B$, llamada dilatación de f por B. Esta nueva función puede calcularse mediante la ecuación (2.2) como $(f \oplus B)(\mathbf{p}) = \max\{a: \mathbf{p} \in T_a(f) \oplus B\}$ o de forma equivalente mediante la fórmula directa

$$(f \oplus B)(\mathbf{p}) = \max\{f(x-y), y \in B\} \tag{2.8}$$

En forma similar, erosionar todos los conjuntos de umbral de f por el mismo conjunto B y superponiendo todas las salidas vía la ecuación (2.2) proporciona una nueva función, la erosión de f por B, que también puede calcularse mediante la siguiente fórmula

$$(f \otimes B)(\mathbf{p}) = \min\{f(x+y), y \in B\} \tag{2.9}$$

La apertura $\circ$ y cierre $\bullet$ de f por B están definidas respectivamente por

$$f \circ B = (f \otimes B) \oplus B \tag{2.10}$$

$$f \bullet B = (f \oplus B) \otimes B \tag{2.11}$$

Los resultados de aplicar los operadores erosión, dilatación, apertura y cierre a una señal discreta cuantizada se muestran en la tabla 2.2 para un elemento estructurante B=[-1, 0, 1]. Note que los puntos finales de las salidas están indeterminados debido a que los desplazamientos en los puntos del elemento estructurante requieren puntos afuera del dominio [0 10]. Una alternativa puede ser asumir algún valor para la señal afuera del intervalo

dado. Es ilustrativo verificar los resultados de la tabla 2.2 al utilizar las ecuaciones 2.8 a 2.11 y al aplicar el conjunto de definiciones teóricas para la representación por umbrales de $f(\mathbf{p})$ en la tabla 2.1.

P	0	1	2	3	4	5	6	7	8	9	10
$f(\mathbf{p})$	1	1	2	1	3	0	0	1	0	2	3
$f(\mathbf{p})\oplus B$	-	2	2	3	3	3	1	1	2	3	-
$f(\mathbf{p})\otimes B$	-	1	1	1	0	0	0	0	0	0	-
$f(\mathbf{p})\circ B$	-	-	2	2	3	1	1	1	1	-	-
$f(\mathbf{p})\bullet B$	-	-	2	2	3	1	1	1	1	-	-

Tabla 2.2. Dilatación, erosión, apertura y cierre de una señal cuantizada discreta.

La figura 2.3 muestra otro conjunto de resultados de aplicar los operadores morfológicos básicos a una señal 1D. En la figura 2.3 observamos que la erosión de una función f por un conjunto pequeño B reduce los picos y alarga los mínimos de la función. La dilatación de f por B incrementa los valles y alarga los máximos de la función. La apertura de B suaviza la gráfica de f desde abajo al cortar sus picos, y el cierre suaviza la gráfica de f desde arriba al llenar sus valles. Es claro que un elemento estructurante 1D grande tenderá a agrandar el efecto de suavizado.

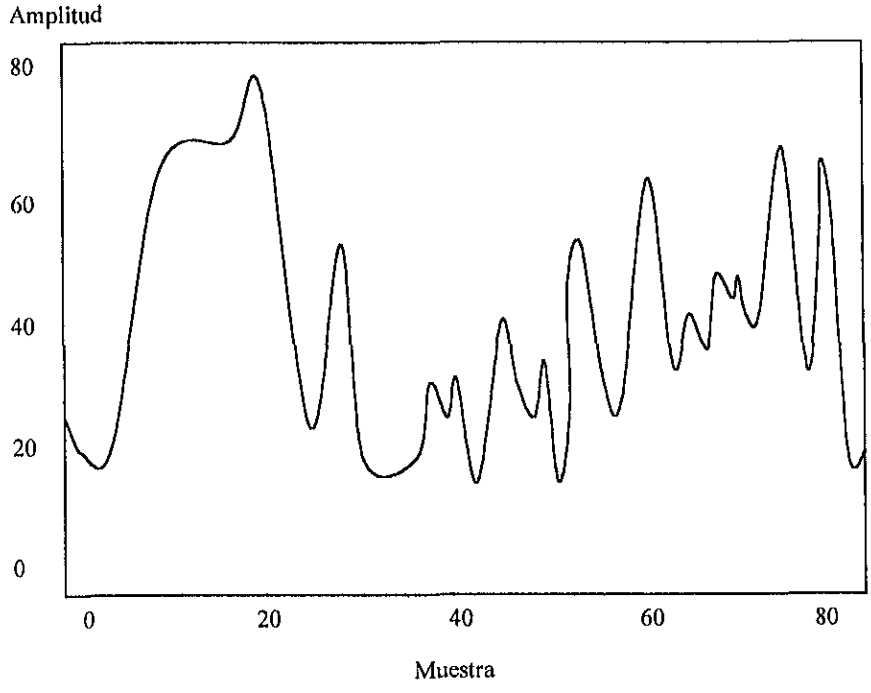
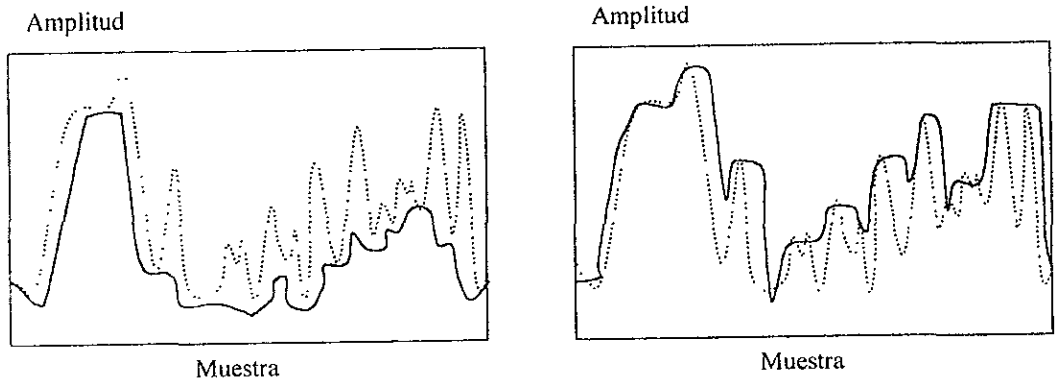
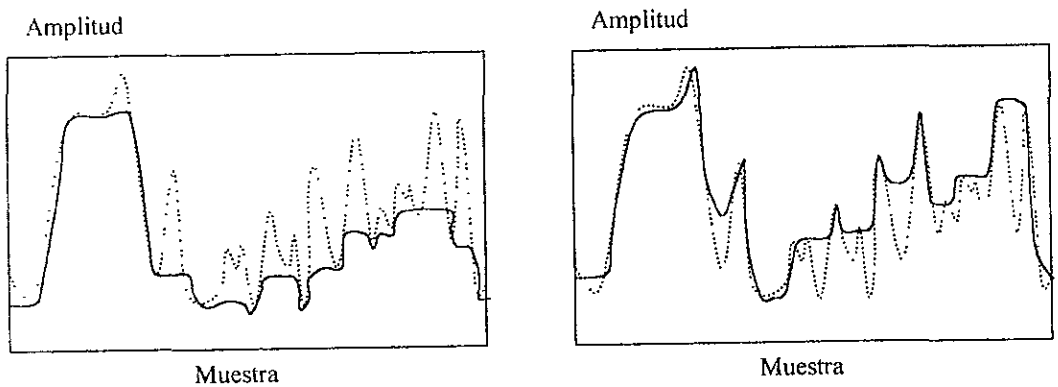


Figura 2.3. a) Función original f.



b) c)

Figura 2.3. b) Erosión, **c)** dilatación de *f* por *B*.



d) e)

Figura 2.3. d) Apertura, **e)** cierre de *f* por *B*.

En general, la erosión $\varepsilon^n(I)$ de la imagen en niveles de grises 2D $I(\mathbf{p})$ $\mathbf{p}=(x, y)$, por un elemento estructurante plano de tamaño NXN=(2n+1)X(2n+1), está dada por

$$\varepsilon^n(I(x, y)) = \min \{I(x - dx, y-dy) \; -n \leq dx \leq n \;\; \text{y} \;\; -n \leq dy \leq n \} \qquad (2.12)$$

En forma similar, la dilatación está dada por

$$\delta^n(I(x, y)) = \max \{I(x - dx, y-dy) \; -n \leq dx \leq n \;\; \text{y} \;\; -n \leq dy \leq n\} \qquad (2.13)$$

La aplicación de la ecuación 2.12 seguida de 2.13 define el operador apertura

$$\gamma^n(I) = \delta^n(\varepsilon^n(I)) \qquad (2.14)$$

De forma similar se define el operador cierre

$$\varphi^n(I) = \varepsilon^n(\delta^n(I)) \qquad (2.15)$$

La aplicación de la ecuación 2.14 seguida de 2.15 define el operador *apertura-cierre* y la aplicación de 2.15 seguida por 2.14 define el operador *cierre-apertura* dados por

$$\varphi^n\gamma^n(I) = \varepsilon^n(\delta^n(\delta^n(\varepsilon^n(I)))) \quad (2.16)$$

$$\gamma^n\varphi^n(I) = \delta^n(\varepsilon^n(\varepsilon^n(\delta^n(I)))) \quad (2.17)$$

respectivamente. Los operadores apertura y cierre remueven detalles brillantes y oscuros respectivamente. Sin embargo el desempeño de éstos operadores es pobre para nuestros propósitos, ya que remueve los bordes de la imagen [28].

En la figura 2.4 mostramos los resultados que se obtienen al filtrar la imagen 2D estándar *Lena* original *I*, a través de los operadores morfológicos básicos, mediante un elemento estructurante plano de 3X3 B=[1, 1, 1; 1, 1, 1; 1, 1, 1].

Figura 2.4. a) Imagen en escala de grises original *I*, *Lena*.

b) **c)**

Figura 2.4. b) Erosión, **c)** dilatación de *I* por *B*.

d) e)

Figura 2.4. d) Apertura, e) cierre de I por B.

Adicionalmente, mostramos en la figura 2.5 los resultados de filtrar I mediante los operadores apertura-cierre y cierre-apertura definidos en las ecuaciones 2.16 y 2.17 utilizando el mismo elemento estructurante B.

a) b)

Figura 2.5. a) Apertura-cierre, b) cierre-apertura de I por B.

2.4. Apertura y cierre por reconstrucción

La reconstrucción morfológica es parte de un conjunto de operadores sobre imágenes frecuentemente conocidos como *geodésicos*. En el caso binario, la reconstrucción simplemente extrae los *componentes conectados* de la imagen

binaria I (la máscara) que está *marcada* por una imagen binaria J contenida en I. La transformación puede extenderse al caso de escala de grises, en donde se convierte en una herramienta extremadamente útil para diversas tareas de análisis de imágenes.

Las características en el proceso de filtrado por reconstrucción son en resumen: filtrado de la imagen, no atenuación de bordes, enfoque geométrico al procesamiento de señales y el tratamiento sencillo con criterios tales como tamaño, forma, contraste y conectividad [20].

La presente sección primero revisa algunas definiciones preliminares para posteriormente incorporarlas al operador *apertura-cierre por reconstrucción.*

2.4.1. Reconstrucción de imágenes binarias

Definición a partir de las componentes conectadas

Tomemos a X y B como dos imágenes binarias definidas en el mismo dominio discreto D tal que $B \subseteq X$. En términos de mapeo, esto significa que $\forall \mathbf{p} \in D$, $B(\mathbf{p})=1 \Rightarrow X(\mathbf{p})=1$. A B se le conoce como imagen marcadora y a X se le conoce como imagen de máscara. Tomemos a X_1, X_2, ..., X_n como los componentes conectados de X.

Definición: La reconstrucción $\rho_X(B)$ de la máscara X a partir de la marca B es la unión de los componentes conectados de X que contienen al menos un pixel de B.

$$\rho_X(B) = \bigcup X_k,\ B \cap X_k \neq 0$$

La definición anterior se ilustra en la figura 2.6. Esta es extremadamente simple pero descubre ciertas aplicaciones de interés.

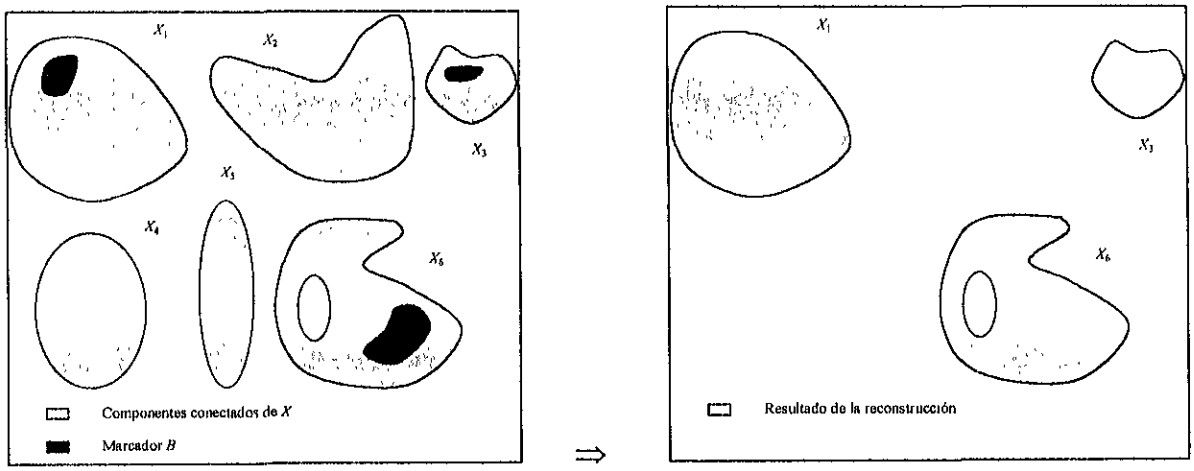


Figura 2.6. Reconstrucción binaria a partir de marcadores.

Definición a partir de la distancia geodésica

La reconstrucción es comúnmente presentada usando la noción de distancia geodésica. Dado un conjunto X (la máscara), la distancia geodésica $d_G(\mathbf{p}, \mathbf{q})$ entre dos pixeles **p** y **q** es la longitud de la ruta más corta que une a **p** y **q** y que está incluida en X. Note que la distancia geodésica entre dos pixeles dentro de una máscara es altamente dependiente del tipo de conectividad que se utiliza. Esta noción es ilustrada en la figura 2.7. En particular, se pueden definir dilataciones geodésicas (y similarmente erosiones geodésicas) como sigue.

Definición: Tomemos a $X \subset \mathbb{Z}^2$ como un conjunto discreto de $\mathbb{Z}^2$ y $Y \subseteq X$. La dilatación geodésica de tamaño $n \geq 0$ de Y dentro de X es el conjunto de pixeles de X cuya distancia geodésica a Y es más pequeña o igual a n

$$\delta_X^{(n)}(Y) = \{\mathbf{p} \in X \mid d_X(\mathbf{p}, Y) \leq n\}$$

A partir de tal definición, es obvio que las dilataciones geodésicas son transformaciones excesivas, ya que $Y \subseteq \delta_X^{(n)}(Y)$. Adicionalmente, la dilatación geodésica de un determinado tamaño n puede obtenerse al iterar n dilataciones geodésicas elementales

$$\delta_X^{(n)}(Y) = \delta_X^{(1)} \cdot \delta_X^{(1)} \ldots \delta_X^{(1)}(Y) \tag{2.18}$$

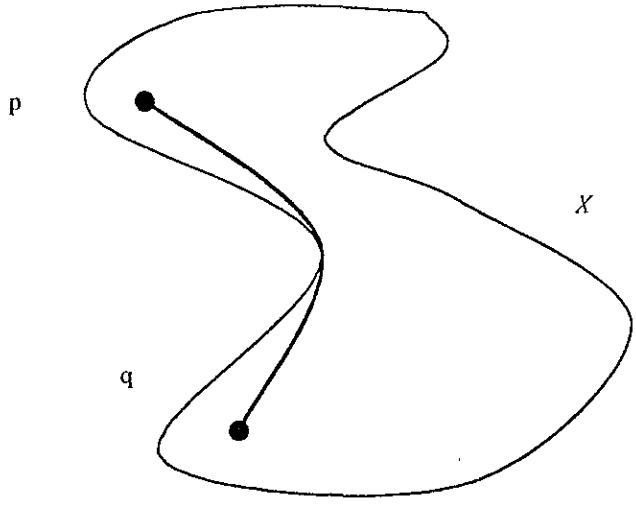


Figura 2.7. Distancia geodésica $d_G(\mathbf{p}, \mathbf{q})$ dentro de X.

La figura 2.8 muestra dilataciones geodésicas sucesivas de un marcador en el interior de una máscara. La dilatación geodésica elemental puede obtenerse a través de una dilatación estándar de tamaño uno seguida de una intersección [28]

$$\delta_X^{(1)} = (Y \oplus B) \cap X \tag{2.19}$$

La ultima ecuación es absolutamente equivocada cuando se consideran dilataciones geodésicas no elementales. Cuando se desarrollan sucesivamente dilataciones geodésicas elementales de un conjunto Y dentro de una máscara

X, las componentes conectadas de X cuya intersección con Y no es vacía son llenadas progresivamente. La siguiente proposición entonces puede plantearse.

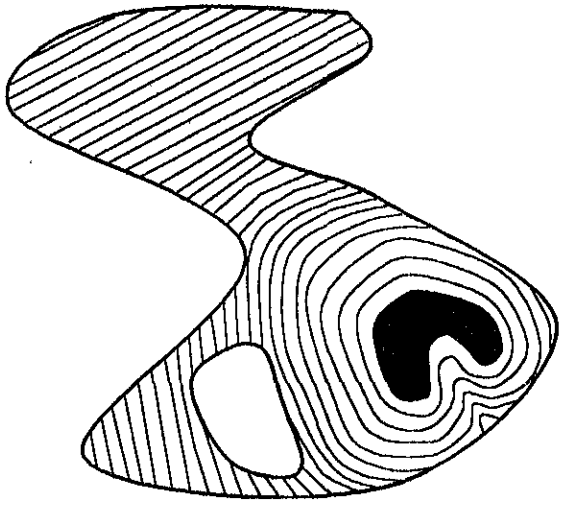


Figura 2.8. Fronteras de dilataciones geodésicas sucesivas de un conjunto dentro de una máscara.

Proposición: La reconstrucción de X a partir de $Y \subseteq X$ se obtiene al iterar dilataciones geodésicas elementales de Y dentro de X hasta la estabilidad. En otras palabras

$$\rho_X(Y) = \bigcup \delta_X^{(n)}(Y), \quad n \geq 1$$

La proposición es la base de uno de los algoritmos más simples para calcular las reconstrucciones geodésicas en ambos casos: binario y escala de grises.

2.4.2. Reconstrucción de imágenes en escala de grises

Definición usando superposición de umbrales

Se conoce de años atrás que -al menos en el caso discreto- cualquier avance en transformaciones definidas para imágenes binarias puede extenderse a imágenes en escala de grises. Por avance entendemos una transformación ψ tal que

$$Y \subseteq X \Rightarrow \psi(Y) \subseteq \psi(X), \quad \forall X, Y \subset Z^2 \tag{2.20}$$

Con el objetivo de extender tal transformación ψ a imágenes en escala de grises I tomando sus valores en $\{0, 1, \ldots, N\text{-}1\}$, es suficiente considerar umbrales sucesivos $T_k(I)$ de I, para k=0 hasta N-1

$$T_k(I) = \{\mathbf{p} \in D_I \mid I(\mathbf{p}) \geq k\}$$

Se dice que éstos constituyen la *descomposición por umbrales* de I. Como se ilustra en la figura 2.9, éstos conjuntos satisfacen la siguiente relación de inclusión

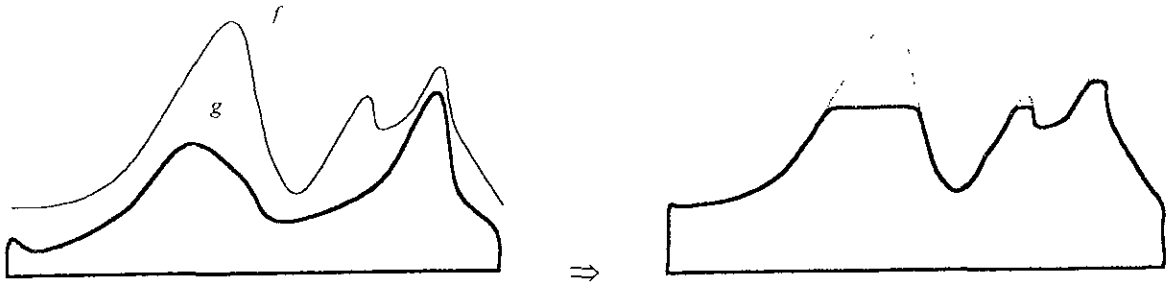


Figura 2.9. Reconstrucción en escala de grises de la máscara f a partir de la marca g.

$$T_k(I) \subseteq T_{k-1}(I),\ \forall k \in [1,\ N\text{-}1]$$

Cuando se aplica la operación ψ avanzada a cada uno de éstos conjuntos, sus relaciones de inclusión se preservan. Entonces podemos extender ψ a imágenes en escala de grises como sigue

$$\forall \mathbf{p} D_I,\quad \psi(I)(\mathbf{p}) = \max\{k \in [0,\ N\text{-}1] \mid \mathbf{p} \in \psi(T_k(I))\} \tag{2.21}$$

En el caso presente, la reconstrucción geodésica binaria es una transformación avanzada en la cual se satisface

$$Y_1 \subseteq Y_2,\ X_1 \subseteq X_2,\ Y_1 \subseteq X_1,\ Y_2 \subseteq X_2,\ \Rightarrow \rho_{X1}(Y_1) \subseteq \rho_{X2}(Y_2) \tag{2.22}$$

Por lo tanto, siguiendo el principio de superposición de umbrales en la ecuación 2.20, definimos la reconstrucción en escala de grises como sigue.

Definición: (Reconstrucción en escala de grises, primera definición): Sea J e I dos imágenes en escala de grises definidas en el mismo dominio, tomando sus valores en el conjunto discreto $\{0,\ 1,\ \ldots,\ N\text{-}1\}$ y tales que $J \leq I$ (por ejemplo, para cada pixel $\mathbf{p} \in D_I$, $J(\mathbf{p}) \leq I(\mathbf{p})$). La reconstrucción en escala de grises $\rho_I(J)$ de I a partir de J está dada por

$$\forall \mathbf{p} \in D_I,\ \rho_I(J)(\mathbf{p}) = \max\{k \in [0,\ N\text{-}1] \mid \mathbf{p} \in \rho_{Tk(I)}(Tk(J))\}$$

La figura 2.9 ilustra ésta información. Justamente igual que la reconstrucción binaria extrae aquellos componentes conectados de la máscara que están marcados, la reconstrucción en escala de grises extrae los picos de la máscara que están marcados por la imagen marcadora. Esta característica es tomada en cuenta en los ejemplos de aplicación mostrados.

Definición alternativa para la reconstrucción en escala de grises

La definición formal no proporciona un método de interés computacional para la determinación la reconstrucción en escala de grises en imágenes digitales. Sin duda, aún si se utiliza un algoritmo para la reconstrucción binaria completamente optimado, se necesita aplicar 256 veces para

determinar la reconstrucción en escala de grises en imágenes de 8 bits. Por lo tanto no es usual introducir ésta transformación utilizando dilataciones geodésicas como se vio anteriormente.

Siguiendo el principio de descomposición en umbrales, se puede definir fácilmente la dilatación geodésica elemental $\delta_I^{(1)}(J)$ de la imagen en escala de grises $J \leq I$ bajo I

$$\delta_I^{(1)}(J)=(J \oplus B) \wedge I \tag{2.23}$$

En ésta ecuación, $\wedge$ significa el mínimo positivo y $J \oplus B$ es la dilatación de J por el elemento estructurante plano B. Estas dos nociones son una extensión directa al caso en escala de grises de la intersección y de la dilatación binaria por B. La dilatación geodésica en escala de grises de tamaño $n \geq 0$ está dada entonces por

$$\delta_I^{(n)}(J)= \delta_I^{(1)} \cdot \delta_I^{(1)} \ldots \delta_I^{(1)} (J) \tag{2.24}$$

Esto conduce entonces a la segunda definición de reconstrucción en escala de grises.

Definición: (Reconstrucción en escala de grises, segunda definición): La reconstrucción en escala de grises $\rho_I(J)$ de I a partir de J se obtiene al iterar dilataciones geodésicas en escala de grises de J bajo I hasta que se alcanza la estabilidad

$$\rho_I(J)=\vee \delta_I^{(n)}(J), \; n \geq 1$$

Resulta fácil verificar que ambas definiciones vistas hasta ahora corresponden a la misma transformación.

De forma similar, la erosión elemental geodésica $\varepsilon_I^{(1)}(J)$ de la imagen en escala de grises $J \geq I$ arriba de I está dada por

$$\varepsilon_I^{(1)}(J)=(J \otimes B) \vee I \tag{2.25}$$

en donde $\vee$ significa el máximo punto y $J \otimes B$ es la erosión de J por el elemento estructurante plano B. La erosión geodésica en escala de grises de tamaño $n \geq 0$ entonces está dada por

$$\varepsilon_I^{(n)}(J)=\varepsilon_I^{(1)} \cdot \varepsilon_I^{(1)} \ldots \varepsilon_I^{(1)}(J) \tag{2.26}$$

Entonces ahora estamos en la posibilidad de definir la reconstrucción dual en escala de grises en términos de la erosión geodésica.

Definición: (Reconstrucción dual): Sean I y J dos imágenes en escala de grises definidas en el mismo dominio D_I tales que $I \leq J$. La reconstrucción dual en escala de grises $\rho_I^*(J)$ de la máscara I a partir del marcador J se obtiene al iterar erosiones geodésicas en escala de grises de J arriba de I hasta que se alcanza la estabilidad

$$\rho_I^*(J)=\wedge \varepsilon_I^{(n)}(J), \quad n \geq 1$$

2.4.3. Filtrado morfológico por reconstrucción

La alternativa para mejorar el desempeño de los filtros morfológicos convencionales presentados en la sección 2.3, es introducir nuevos operadores morfológicos basados en erosión geodésica y dilatación geodésica, que están definidos por

$$\varepsilon_R^{(1)}(I)(x, y) = \max \{\varepsilon^{(1)}(I)(x, y), R(x, y)\} \quad (2.27)$$

$$\delta_R^{(1)}(I)(x, y) = \min \{\delta^{(1)}(I)(x, y), R(x, y)\} \quad (2.28)$$

En donde R es una imagen de referencia. Aplicando las ecuaciones 2.27 y 2.28 hasta la estabilidad se definen, respectivamente, la *erosión por reconstrucción* y la *dilatación por reconstrucción*

$$\varepsilon_R^{(rec)}(I) = \varepsilon_R^{(1)}(\cdots \varepsilon_R^{(1)}(\varepsilon_R^{(1)}(I)) \cdots) \quad (2.29)$$

$$\delta_R^{(rec)}(I) = \delta_R^{(1)}(\cdots \delta_R^{(1)}(\delta_R^{(1)}(I)) \cdots) \quad (2.30)$$

En base a éstos operadores se definen los operadores *apertura por reconstrucción de la erosión* y *cierre por reconstrucción de la dilatación*

$$\gamma^{(rec)(n)}(I) = \delta_I^{(rec)}(\varepsilon^{(n)}(I)) \quad (2.31)$$

$$\varphi^{(rec)(n)}(I) = \varepsilon_I^{(rec)}(\delta^{(n)}(I)) \quad (2.32)$$

De igual forma que la *apertura y cierre*, la *apertura por reconstrucción de la erosión* y el *cierre por reconstrucción de la dilatación*, respectivamente, eliminan detalles brillantes y oscuros sin corromper los bordes. Entonces la aplicación de la ecuación 2.31 seguida de 2.32 define el operador *apertura-cierre por reconstrucción*

$$\varphi\gamma^{(rec)(n)}(I) = \varphi^{(rec)(n)}(\gamma^{(rec)(n)}(I)) \quad (2.33)$$

La aplicación de la ecuación 2.33 sobre la imagen original resulta en una imagen simplificada en donde se eliminan los detalles brillantes y oscuros ya que la reconstrucción restablece los contornos que no se han removido totalmente. Por lo tanto se eliminan los detalles perceptualmente menos significativos. La figura 2.10 muestra *Lena* original y después de la transformación por reconstrucción indicada en la ecuación 2.33 con un elemento estructurante de tamaño 3X3, en donde se puede observar el trabajo de conexión de regiones planas realizado por el filtro por reconstrucción.

a) b)

Figura 2.10. a) *Lena* original, **b)** *Lena* después del filtrado por apertura cierre por reconstrucción.

2.5. Comparación entre filtros morfológicos

La figura 2.11 muestra los detalles (64, 149) a (100, 185) ampliados de las imágenes *Lena* original, *Lena* filtrada por apertura cierre, y *Lena* filtrada por apertura cierre por reconstrucción.

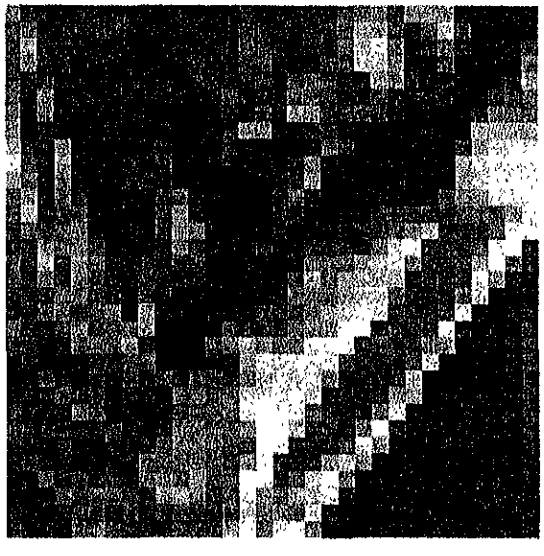
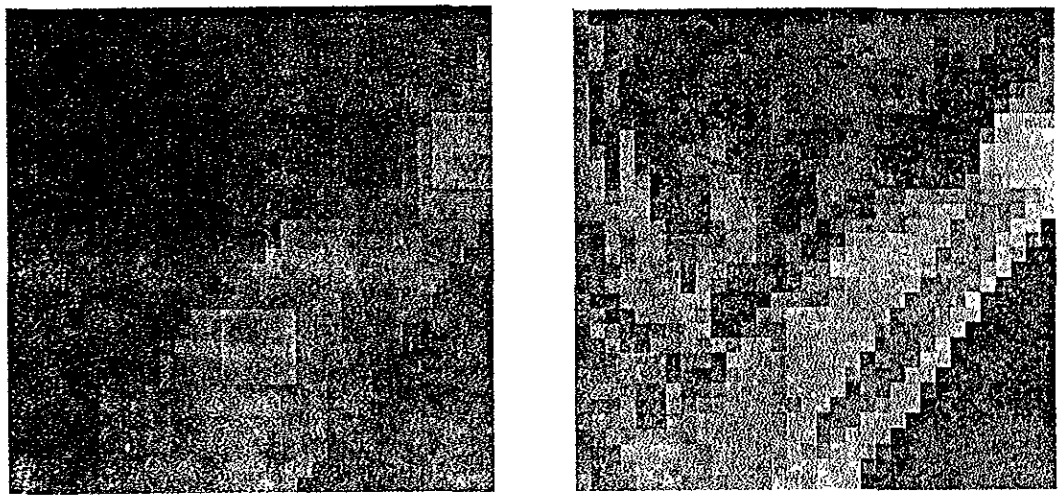


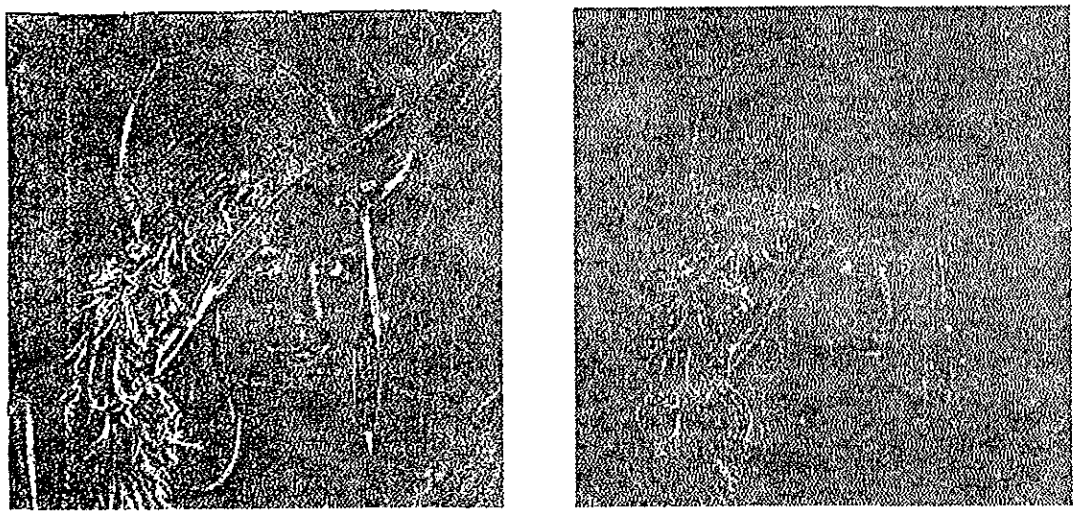
Figura 2.11. a) Detalle de *Lena* original, (64, 149) a (100, 185).



b) c)

Figura 2.11. b) detalle de *Lena* después del filtrado apertura-cierre **c)** detalle de *Lena* después del filtrado apertura-cierre por reconstrucción, (64, 149) a (100, 185)

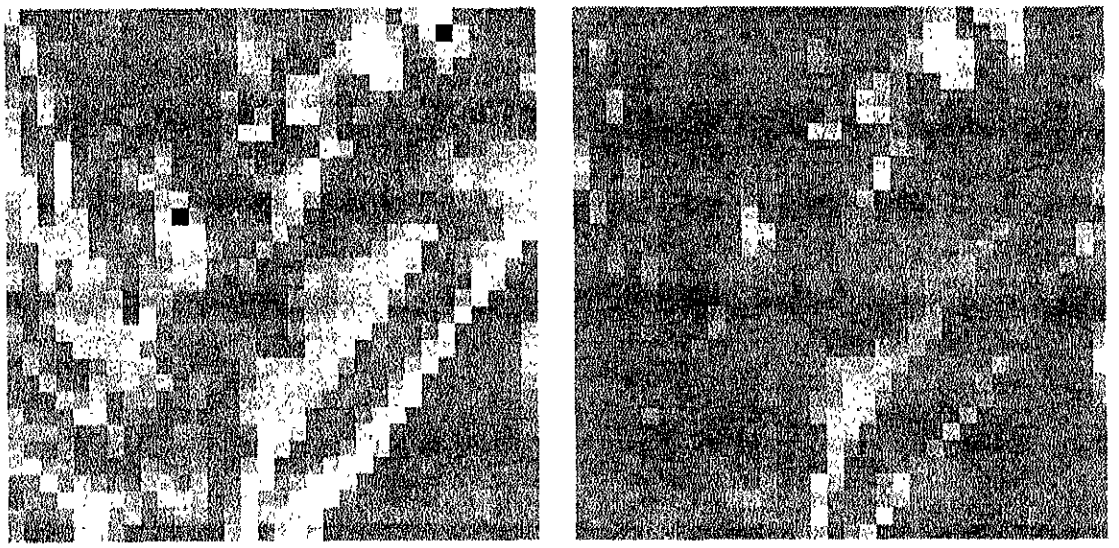
La imagen de diferencias entre los filtros morfológicos convencional y por reconstrucción de las figuras 2.5.a y 2.10, se muestra en la figura 2.12. La figura 2.12 nos proporciona una idea gráfica acerca del desempeño a partir de las regiones planas después del filtrado, así como de los bordes que aún permanecen sin remover.



a) b)

Figura 2.12. a) Imagen de diferencias entre *Lena* original y *Lena* después del filtrado apertura-cierre, **b)** Imagen de diferencias entre *Lena* original y *Lena* después del filtrado apertura-cierre por reconstrucción.

Finalmente la figura 2.13 muestra los detalles de las imágenes de diferencias a partir de las figuras en 2.12.



a) b)

Figura 2.13. a) Detalle de la diferencia entre *Lena* original y *Lena* después del filtrado apertura-cierre, **b)** Detalle de la diferencia entre *Lena* original y *Lena* después del filtrado apertura-cierre por reconstrucción.

Capitulo 3

Segmentación espacial

3.1. Generalidades

En la mayoría de las aplicaciones, el primer paso del análisis de imágenes consiste generalmente en segmentar la imagen, como se muestra en la figura 3.1. La segmentación subdivide una imagen en sus partes constituyentes u objetos. Para la visión por computadora el problema de segmentar no es tan trivial como lo es para la visión humana. El nivel de subdivisión depende del problema a resolver en visión por computadora y del detalle deseado por el observador en la visión humana.

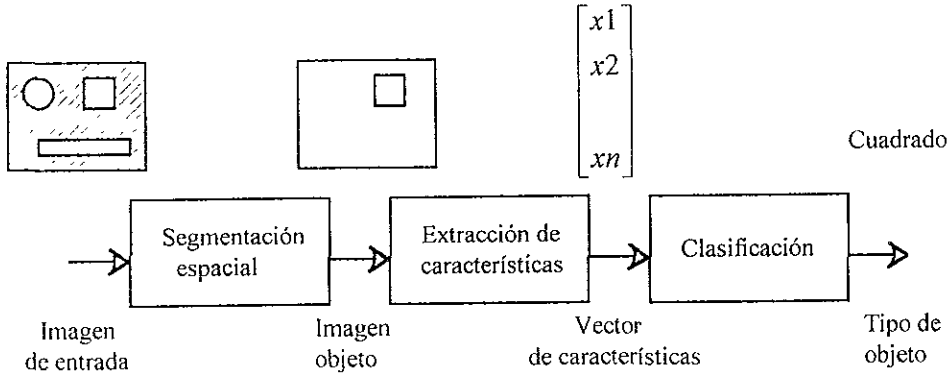


Figura 3.1. Las tres etapas en visión por computadora.

Centrando nuestra atención en la visión por computadora, el proceso de segmentación deberá aislar los objetos que sean de interés para una aplicación particular. Por ejemplo, en las aplicaciones automáticas de identificación de vehículos en una carretera, el primer paso consiste en segmentar la carretera de la imagen y a continuación segmentar el contenido de la carretera. En éste sentido, solamente tiene sentido aislar los objetos cuyas características sean similares a las de un vehículo, siendo innecesario extender el proceso de segmentación a detalles perceptualmente menores. No tiene objeto llevar la segmentación por debajo de la escala establecida, ni tampoco hay necesidad de segmentar la imagen para componentes fuera de la carretera.

En general, la segmentación autónoma es una de las tareas más difíciles del procesamiento de imágenes. De hecho, la segmentación rara vez llega a alcanzar niveles satisfactorios. Por esta razón, en ciertas aplicaciones que así lo permiten, se han adoptado dos enfoques que simplifican el problema:

a) Establecer restricciones en los ambientes físicos que rodean a la imagen. En algunas situaciones, tales como aplicaciones de inspección industrial, es posible que se tomen medidas de control sobre el entorno. Es decir, se pueden establecer ambientes de iluminación apropiados, colores contrastantes que identifiquen a los objetos de interés, distancias predeterminadas entre el objetivo y el dispositivo generador de la imagen, sensores que permitan eliminar información irrelevante, mecanismos auxiliares de enfoque y transporte de objetivos, etc.

b) Establecer hipótesis considerando un conocimiento previo de la imagen. En ciertas áreas de aplicación, el especialista del área puede establecer restricciones que pueden simplificar el problema para el especialista en imágenes. Por ejemplo, en procesamiento de imágenes médicas, el médico especialista puede inferir a priori ciertas propiedades acerca de la imagen como umbrales de comparación, estadística, contenido espectral, textura, etc.

Restringiendo aún más las condiciones para la simplificación, en nuestro trabajo utilizamos algoritmos de segmentación de imágenes en 256 niveles de gris. También bajo ésta suposición, los algoritmos de segmentación de imágenes monocromáticas generalmente se basan en una de las dos propiedades básicas de los valores de nivel de gris: discontinuidad y similitud. En la primera categoría, el método consiste en dividir una imagen basándose en los cambios bruscos de nivel de gris. Las principales áreas de interés de esta categoría son la detección de puntos aislados y la detección de bordes en una imagen. Los principales métodos de la segunda categoría están basados en la umbralización, crecimiento de región, y división y fusión de regiones. El concepto de segmentación basado en la discontinuidad o similitud de los valores de nivel de gris de sus pixeles es aplicable tanto a las imágenes estáticas (segmentación espacial) como a las dinámicas (segmentación espacio-temporal). En el último caso, el movimiento puede utilizarse a menudo como un poderoso indicador para mejorar el rendimiento de los algoritmos de segmentación (capítulo cuatro).

El capítulo está organizado como se describe a continuación. La segunda sección revisa técnicas fundamentales para la umbralización que son utilizadas como herramientas cotidianas en el procesamiento de imágenes. La tercera sección contiene técnicas de segmentación basadas en detección de discontinuidades, también conocidas como técnicas de segmentación basadas en gradientes. La cuarta sección incorpora técnicas de segmentación basadas en regiones, las cuales mejoran el desempeño de la técnicas anteriores al considerar que las imágenes están formadas perceptualmente por regiones disjuntas.

3.2. Umbralización

La umbralización es una técnica muy útil en particular para escenas que contienen objetos sólidos contrastando fuertemente con el fondo. Es computacionalmente simple y nunca falla al definir regiones.

Cuando se utiliza una regla de umbralización para segmentación de imágenes, uno asigna a todos los pixeles arriba del umbral un nivel de gris del objeto. Todos los pixeles con nivel de gris abajo del umbral definen las partes ajenas al objeto. La frontera entonces resulta ser el conjunto de los puntos interiores al objeto, cada uno de los cuales tiene al menos un vecino de la región exterior.

La umbralización funciona bien si los objetos de interés tienen nivel de gris uniforme y el resto del fondo diferente nivel de gris, pero también uniforme. Si los objetos difieren del fondo en alguna otra propiedad diferente al nivel de gris (textura, etc.), se puede utilizar alguna operación que convierta esa propiedad al nivel de gris. Entonces la umbralización por nivel de gris puede segmentar una imagen preprocesada.

3.2.1. Umbralización global

En la más simple implementación de la localización de fronteras por umbralización, el valor del umbral de comparación se mantiene constante a través de la imagen. Si el nivel de gris en el fondo es uniforme, y si los objetos tienen aproximadamente igual contraste diferente del fondo, entonces usualmente un nivel fijo de umbral global funcionará bien, tomando en cuenta que se ha elegido apropiadamente el nivel de gris en el umbral; figura 3.2.

a) b)

Figura 3.2. a). Imagen original en nivel de gris. b). Imagen umbralizada.

3.2.2. Umbralización adaptable

En muchos casos, el nivel de gris del fondo no es constante, y el contraste con los objetos varia a lo largo de la imagen. En tales casos, un umbral que funcione apropiadamente en algunas regiones de la imagen no funcionará para el resto. En éstos casos, es conveniente utilizar un nivel de gris de umbral que varíe con la posición de la imagen.

La figura 3.3 muestra una imagen digitalizada de un plano mecánico. En ésta imagen, el nivel de gris en el fondo varia debido a la iluminación no uniforme que proporciona el dispositivo de digitalización (escáner). Por lo tanto, el contraste varía para diferentes zonas de la imagen. En la figura 3.3.a), se ha utilizado una umbralización global con nivel constante, con el objeto de aislar los detalles en el dibujo. En la figura 3.3.b), el umbral fue variado para diferentes áreas de la imagen utilizando un contraste local. El resultado último produce menos errores de segmentación, principalmente en casos donde existen múltiples caracteres agrupados. Un estudio similar muestra que la exactitud de las mediciones de las áreas de los objetos de interés fue mejorada al utilizar umbralización adaptable. En la figura 3.3.b), el umbral para cada región fue especificado aproximadamente por la parte media entre el nivel de gris del interior contra el nivel de gris del exterior local.

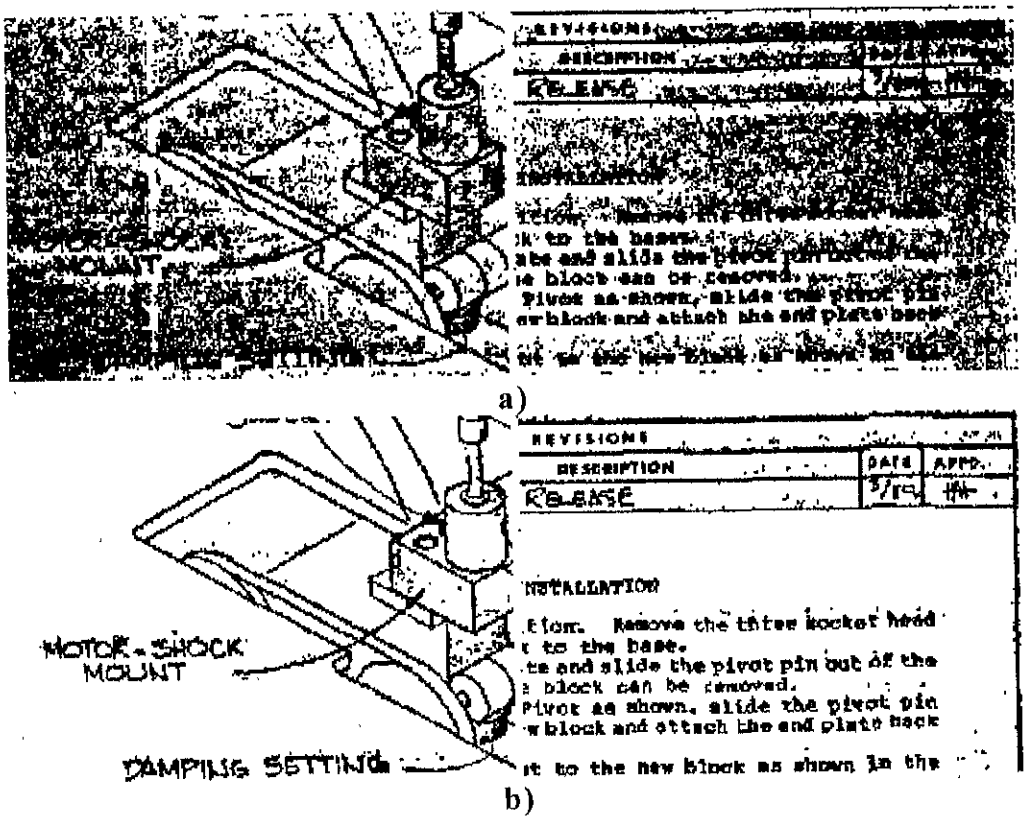


Figura 3.3. a). Umbralización global. **b).** Umbralización adaptable.

3.2.3. Umbralización óptima

A menos que el objeto en la imagen tenga extremadamente alto contraste contra el fondo, el valor exacto del nivel de gris del umbral puede tener un considerable efecto en la posición de las fronteras y el tamaño total del objeto extraído. Esto significa que las mediciones subsecuentes de tamaño, particularmente área, son sensibles al nivel de umbral. Por esta razón, se necesita un método óptimo, o al menos consistente, para establecer el umbral.

Técnicas basadas en histograma

Una imagen que contiene un objeto contrastando con el fondo tiene un histograma bimodal como el mostrado en la figura 3.4, en donde D es el nivel de gris y H es el número de ocurrencias. Los dos picos corresponden al relativamente alto número de pixeles en el interior y exterior del objeto, respectivamente. La inmersión entre los dos picos corresponde al relativamente menor número de artefactos. En casos como éste, el histograma es comúnmente usado para establecer el nivel de gris en el umbral.

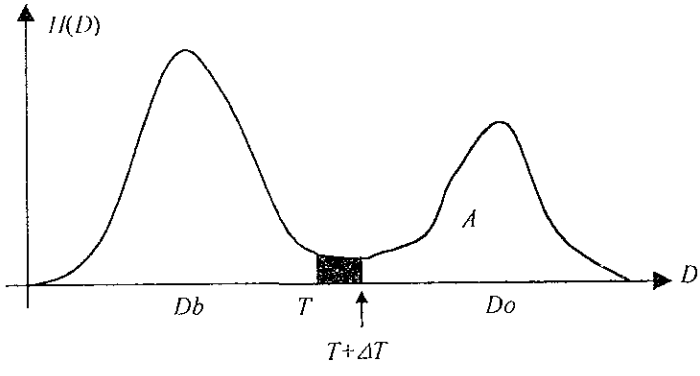


Figura 3.4. Histograma bimodal.

El área de un objeto definida por el nivel de gris del umbral T es

$$A = \int_T^{\infty} H(D)dD$$

Note que al incrementar el umbral de T a $T+\Delta T$ origina sólo un ligero decremento en área si el umbral corresponde a la inmersión del histograma. Por lo tanto, poner el umbral en la inmersión minimiza la sensibilidad a las mediciones de área a los errores pequeños en la selección del umbral.

Si la imagen o región de la imagen que contiene el objeto es ruidosa y no muy grande, el histograma por sí mismo será ruidoso. A menos que la inmersión sea aguda, el ruido hará esta localidad obscura, o al menos no localizable claramente. Esto puede ser solucionado en alguna medida al suavizar el histograma, utilizando ya sea procedimientos de convolución o ajuste a curva. Si los dos picos son desiguales en tamaño, el suavizado tenderá a desplazar la posición del mínimo. Los picos, sin embargo, son fácilmente localizables y relativamente estables bajo niveles razonables de suavizado. Un

método más realizable es poner el umbral en una posición fija relativa a los dos picos, por ejemplo en la parte media. Los dos picos representan los niveles de gris del objeto y su exterior. En general, estos parámetros pueden ser estimados más realmente que la inmersión sobre el histograma.

Uno puede formar un histograma de sólo aquellos pixeles que tienen relativamente altas magnitudes de gradiente, por ejemplo los mayores al 10%. Esto elimina el gran número de pixeles exteriores e interiores bajo consideración y puede hacer que la inmersión del histograma sea más accesible. También se puede dividir el histograma mediante el gradiente promedio de los pixeles en cada número de nivel de gris, para aumentar la inmersión, o promediar el nivel de gris de los pixeles con alto gradiente para determinar un umbral. Por otra parte se puede dividir el histograma de manera que las partes divididas preserven su estadística o algún criterio con respecto a los momentos estadísticos.

El filtro laplaciano es un derivador de segundo orden. El filtrado laplaciano, seguido de un suavizado y umbralización a un nivel de gris cero o ligeramente mayor, tiende a segmentar objetos en los cruces por cero de la segunda derivada, que corresponden a los puntos de inflexión ocasionados por las fronteras de los objetos. Adicionalmente, el histograma en dos dimensiones del nivel de gris contra el gradiente, puede utilizarse para establecer el criterio de segmentación.

3.3. Segmentación basada en detección de discontinuidades

Existen tres tipos básicos de discontinuidades de una imagen digital: *puntos*, *líneas* y *bordes*. En la práctica, la forma más común de detectar discontinuidades es pasar una máscara (similar al elemento estructurante) a través de la imagen en la forma descrita por la figura 2.1. Para una máscara de 3X3 como la que se muestra en la figura 3.5, el procedimiento implica calcular la suma de los productos de los coeficientes por los niveles de gris contenidos en la región encerrada por la máscara. Esto es, la respuesta de la máscara en un punto cualquiera de la imagen es

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Figura 3.5. Máscara general de 3X3.

$$R = \sum_{i=1}^{9} w_i z_i \tag{3.1}$$

donde z_i es el nivel de gris asociado con el coeficiente de la máscara w_i. Entonces la respuesta de la máscara R está definida con respecto a la posición de su centro. Cuando la máscara está centrada en un pixel del límite, la respuesta se calcula utilizando el entorno parcial apropiado.

3.3.1. Detección de puntos

La detección de puntos aislados de una imagen se realiza utilizando la máscara de la figura 3.6. Se sabe que se ha detectado un punto en la posición central de la máscara si

-1	-1	-1
-1	8	-1
-1	-1	-1

Figura 3.6. Máscara de 3X3 para detectar puntos aislados.

$$|R| > T \tag{3.2}$$

en donde T es un umbral no negativo y R está dado por la ecuación 3.1. Básicamente, lo que hace tal esquema es medir las diferencias ponderadas entre el punto central y sus vecinos, partiendo de la idea de que el nivel de gris de un punto aislado será bastante distinto al de sus vecinos.

La máscara de la figura 3.6 es la misma que se utiliza para los filtros de alta frecuencia espacial, en son de interés las diferencias lo suficientemente grandes (que determina T) para que se consideren puntos aislados en la imagen.

3.3.2. Detección de líneas

El siguiente nivel de complejidad implica la detección de líneas en una imagen. Considere la máscara que se muestra en la figura 3.7. Si la primera máscara se trasladara por toda una imagen, podría tener un resultado más significativo a líneas orientadas horizontalmente. Con un fondo constante, la respuesta máxima resultará cuando la línea pase por la fila central de la máscara. Esto se comprueba fácilmente esbozando una simple matriz de unos, con una línea de gris diferente (por ejemplo 5) recorriendo horizontalmente la matriz. Un experimento similar puede verificar que la segunda máscara de la figura 3.7 responde mejor a las líneas orientadas a 45°; la tercera a las líneas verticales y la cuarta a las líneas en dirección de -45°. Estas direcciones

también pueden establecerse al observar que la dirección preferida de cada máscara está ponderada por un coeficiente mayor que las otras direcciones posibles.

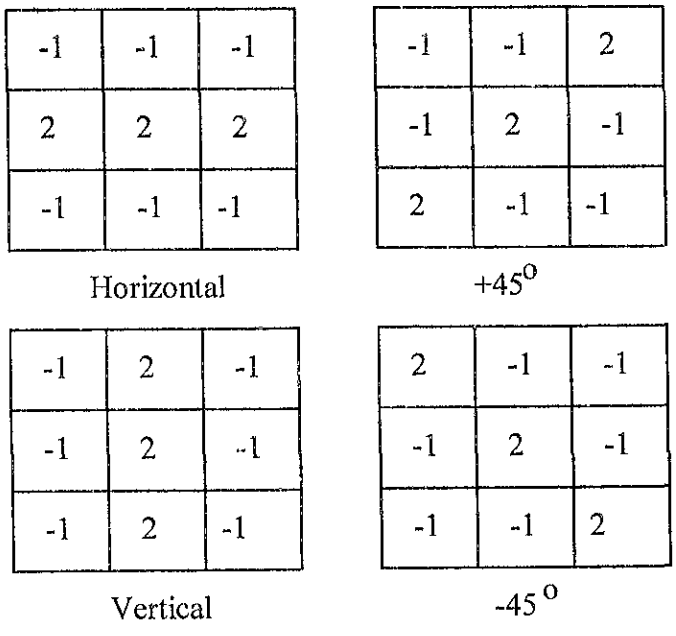


Figura 3.7. Máscaras de 3X3 para detectar líneas.

Sean R_1, R_2, R_3, y R_4 las respuestas de las máscaras de la figura 3.7, de izquierda a derecha y arriba a abajo, donde R_i esta dado por la ecuación 3.1. Suponga que todas las máscaras pasan por una imagen. Si en cierto punto de la imagen $|R_i| > |R_j|$, para todo $j \neq i$, éste punto será el que tenga mayor probabilidad de estar asociado con una línea en la dirección de la máscara y. Por ejemplo, si en un punto de la imagen, $|R_1| > |R_j|$, para j=2, 3, 4, éste punto en particular será el que tenga mayor probabilidad de estar asociado con una línea horizontal.

3.3.3. Detección de bordes

Aunque la detección de punto y línea son evidentemente herramientas útiles, la detección de bordes es por mucho el método más común para identificar discontinuidades en imágenes en nivel de gris. La razón es que los puntos aislados y las líneas delgadas no son de frecuente preocupación en la mayor parte de las aplicaciones prácticas.

Un borde es la frontera entre dos regiones con propiedades de nivel de gris relativamente distintas. La técnica supone que las regiones en cuestión son lo suficientemente homogéneas para que la transición entre dos de ellas se pueda determinar sobre la base de las discontinuidades de nivel de gris solamente. Cuando esta suposición no es valida, las técnicas de segmentación expuestas en la siguiente sección son normalmente de mayor aplicación que la detección de bordes.

Básicamente la idea que subyace en la mayor parte de las técnicas de detección de bordes es el cálculo de un operador local de derivación. La figura 3.8 ilustra el concepto. La figura 3.8. a) muestra una imagen de una banda clara sobre un fondo oscuro, el perfil del nivel de gris a lo largo de una línea de exploración horizontal de la imagen, y la primera y segunda derivadas del perfil. Se observa que en el perfil un borde está modelado como un cambio suave del nivel de gris. Este modelo refleja el hecho que los bordes de las imágenes digitales están generalmente emborronados a causa del dispositivo de adquisición.

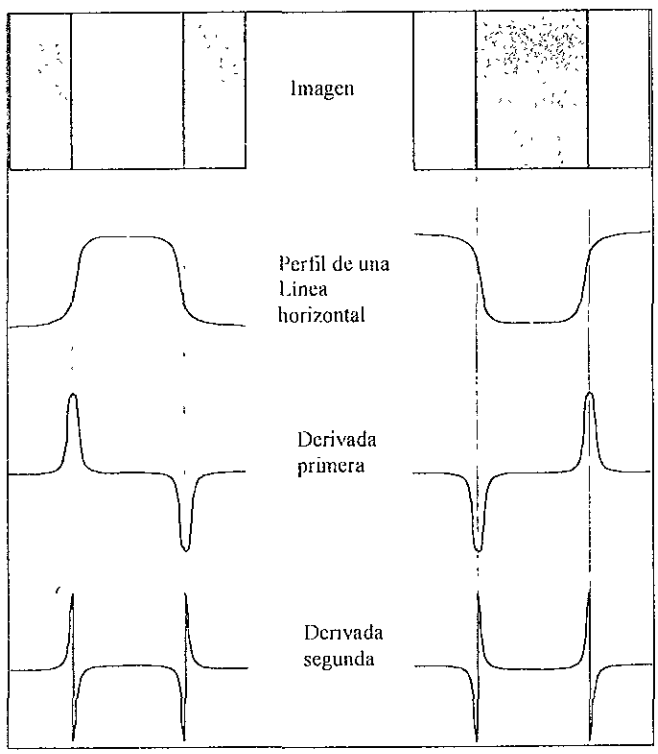


Figura 3.8. Detección de bordes por operadores de derivación. **a).** Banda clara sobre fondo oscuro. **b).** Banda oscura sobre fondo claro.

La figura 3.8 a) muestra que la primera derivada del perfil de nivel de gris es positiva en el borde de subida de la transición, negativa en el borde de salida y cero en las zonas de nivel de gris constante. La derivada segunda es positiva en la parte de la transición asociada con el lado oscuro del borde, negativa en la parte de la transición asociada con el lado claro y cero en las zonas de nivel de gris constante. Por lo tanto, el módulo de la derivada primera se puede utilizar para detectar la presencia de un borde en una imagen, y el signo de la derivada segunda se puede utilizar para determinar si un pixel borde está situado en el lado oscuro o claro del mismo. Se observa que la derivada segunda tiene un paso por cero en el punto medio de una transición del nivel de gris. Como se mostrará más tarde, los pasos por cero proporcionan un poderoso método para localizar bordes en una imagen.

Si bien la presentación anterior se ha limitado al caso unidimensional, se puede aplicar un argumento similar a un borde de cualquier orientación de una

imagen. Simplemente se define un perfil perpendicular a la dirección del borde en cualquier punto que se desee y se interpretan los resultados como en la presentación anterior. La derivada primera en un punto de una imagen se obtiene utilizando el módulo del gradiente en ese punto. La derivada segunda se obtiene de forma similar utilizando el laplaciano.

Operadores gradiente

Se sabe que el concepto del gradiente se puede utilizar para la diferenciación de imágenes. El gradiente de una imagen $f(x, y)$ en la posición (x, y) es el vector

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix} \tag{3.3}$$

Se sabe del análisis vectorial que el gradiente de un vector indica la dirección de la máxima variación de f en (x, y). Una importante cantidad en la detección de bordes es el módulo de éste vector, al que generalmente se le llama, por simplicidad, con la notación ∇f, donde

$$\nabla f = \text{mag}\{\nabla f\} = [G_x^2 + G_y^2]^{1/2} \tag{3.4}$$

Esta cantidad es igual a la máxima variación de $f(x, y)$ por unidad de distancia en la dirección $\nabla \mathbf{f}$. Es práctica común aproximar el gradiente por sus valores absolutos

$$\nabla f \approx |G_x| + |G_y| \tag{3.5}$$

que son más fáciles de implementar, particularmente con hardware dedicado.

La dirección del vector gradiente es también una cantidad importante. Sea $\alpha(x, y)$ la representación del ángulo de dirección del vector $\nabla \mathbf{f}$ en (x, y). Entonces, del análisis vectorial

$$\alpha(x,y) = \tan^{-1}\left(\frac{G_y}{G_x}\right) \tag{3.6}$$

donde el ángulo se mide con respecto al eje x.

De las ecuaciones 3.3 y 3.4 se deduce que el cálculo del gradiente de una imagen se basa en la obtención de las derivadas parciales

$$\frac{\partial f}{\partial x} \quad \frac{\partial f}{\partial y}$$

en cada posición de pixel. Las derivadas se pueden implementar de diversas formas. Sin embargo, los operadores de Sobel tienen la ventaja de proporcionar tanto una diferenciación como un efecto de suavizado. Como las derivadas realzan el ruido, éste efecto de suavizado es una característica

particularmente atractiva de los operadores de Sobel. En la figura 3.9, las derivadas basadas en las máscaras del operador de Sobel son

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \tag{3.7}$$

y

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \tag{3.8}$$

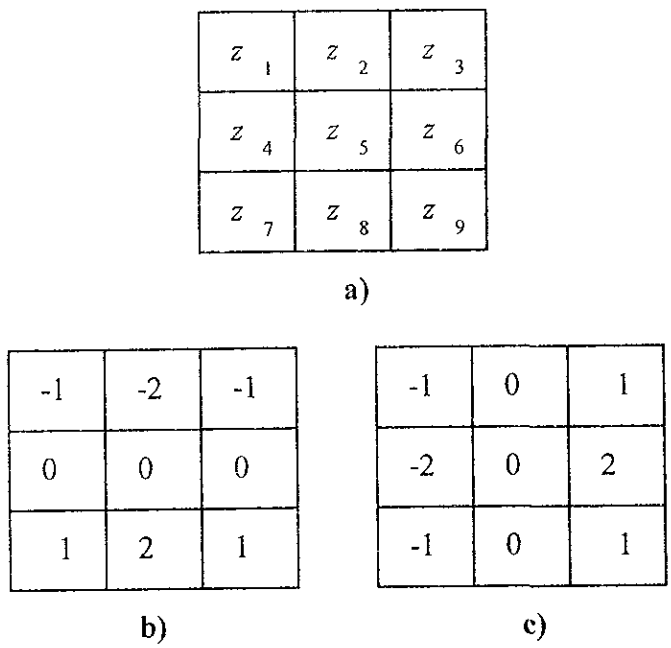


Figura 3.9. Máscaras para los operadores de Sobel. **a).** Región imagen. **b).** Región para calcular G_x. **c).** Región para calcular G_y.

donde z_i son los niveles de gris de los pixeles solapados por las máscaras en cualquier posición de la imagen. Entonces se calcula el gradiente en la ubicación del centro de las máscaras utilizando las ecuaciones 3.4 o 3.5 que proporcionan un valor del gradiente. Para obtener el valor siguiente, se desplazan las máscaras a la posición del siguiente pixel y se repite el proceso. Por lo tanto, una vez que se ha completado el proceso en todas las posibles ubicaciones, el resultado es una imagen gradiente del mismo tamaño que la imagen original. Las operaciones de máscara en los límites de una imagen se implementan utilizando entornos parciales.

La figura 3.10 muestra un ejemplo en donde se aplican los conceptos descritos.

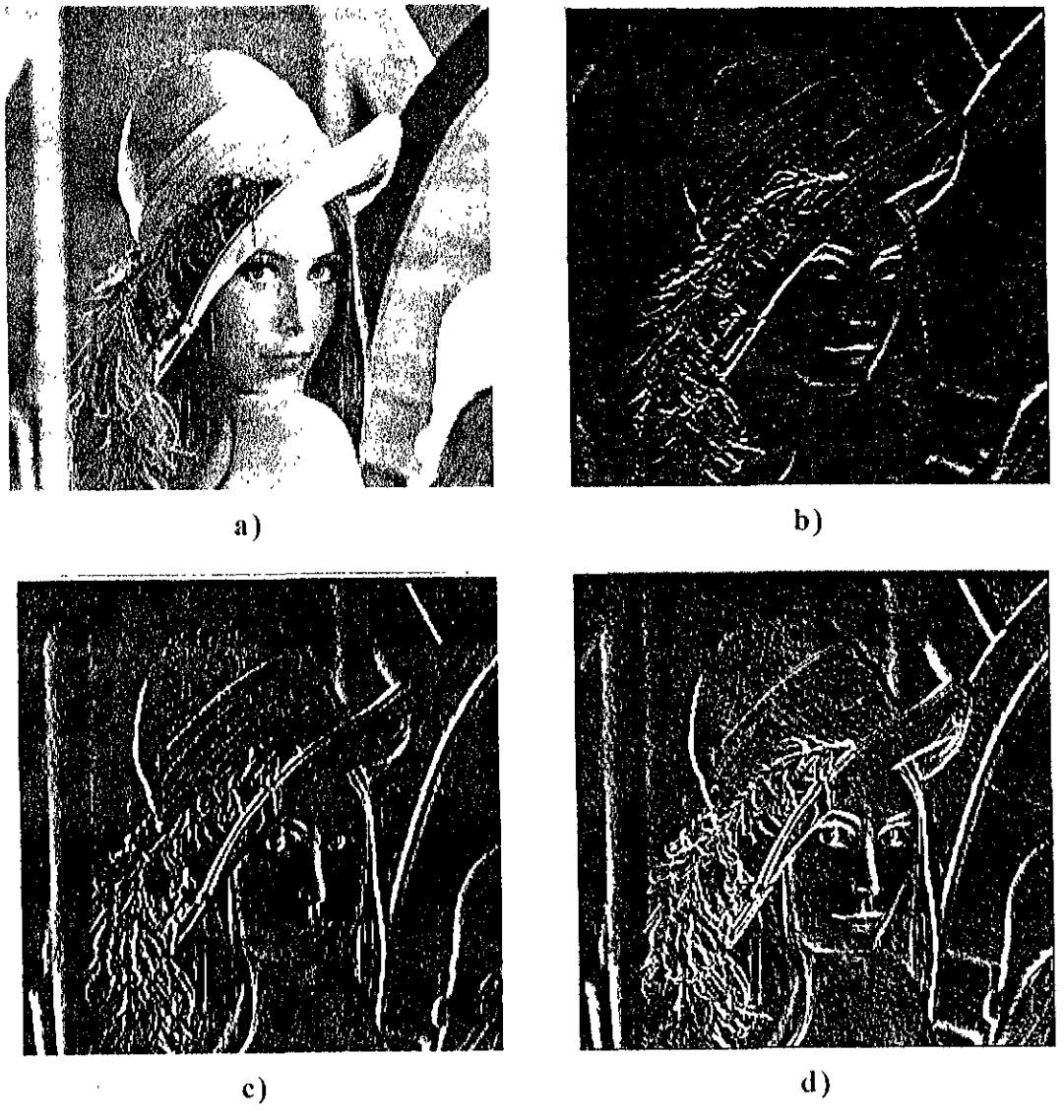


Figura 3.10. a). Imagen original. **b).** Obtención de G_x. **c).** Obtención de G_y. **d).** Imagen completa al utilizar la ecuación 3.5.

Operador Laplaciano

El laplaciano de una función bidimensional $f(x, y)$ es una derivada de segundo orden definida por.

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{3.9}$$

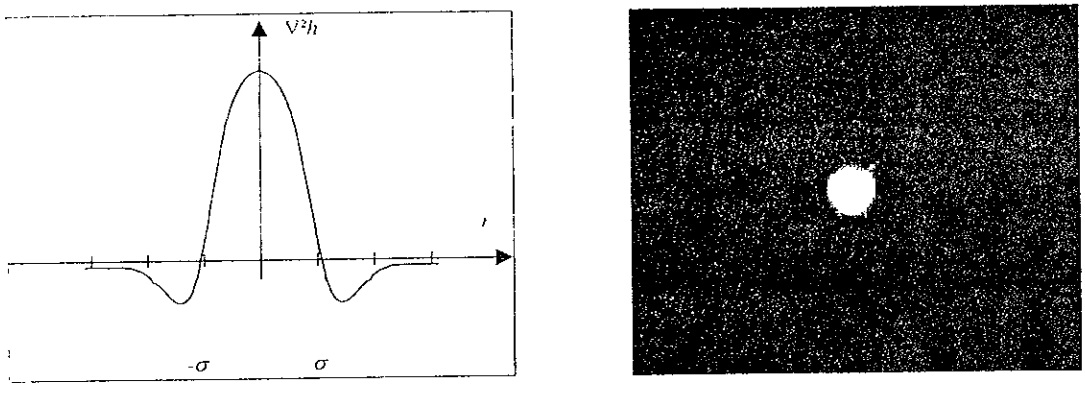
Como en el caso del gradiente, la ecuación 3.9 puede implementarse de forma discreta de varias maneras. Para una región de 3X3, la forma que se encuentra frecuentemente en la práctica es

$$\nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8) \tag{3.10}$$

donde las z corresponden a los valores de la imagen. El requisito básico para la definición del laplaciano digital es que el coeficiente asociado con el pixel central sea positivo y los coeficientes asociados con los pixeles exteriores sean negativos, como lo muestra la figura 3.12. a). Como el laplaciano es una derivada, la suma de los coeficientes debe ser cero. En consecuencia la respuesta es cero siempre que el punto en cuestión y sus vecinos tengan el mismo valor. La figura 3.11 muestra una máscara espacial que se puede utilizar para implementar la ecuación 3.10.

0	-1	0
-1	4	-1
0	-1	0

Figura 3.11. Máscara de 3X3 para calcular el laplaciano.



a) b)

Figura 3.12. a). Sección transversal de $\nabla^2 h$. b)$\nabla^2 h$ como función de la intensidad.

Si bien el laplaciano responde a las transiciones de intensidad, rara vez se utiliza en la práctica para la detección de bordes, por varias razones. Como es una derivada de segundo orden normalmente es sensible al ruido. Además produce bordes dobles (figura 3.8) y es incapaz de detectar direcciones de borde. Por estas razones, el laplaciano desempeña habitualmente un papel secundario de detección para establecer si un pixel está en la parte clara u oscura de un borde.

Un empleo más general del laplaciano consiste en encontrar la ubicación de bordes utilizando sus propiedades de paso por cero (figura 3.8). Este concepto está basado en la convolución de una imagen con el laplaciano de una función gaussiana bidimensional de la forma

$$h(x,y) = \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \qquad (3.11)$$

donde σ es la desviación estándar. Sea $r^2 = x^2 + y^2$. Entonces de la ecuación 3.9 el laplaciano de h, es decir la derivada segunda de h con respecto a r, es

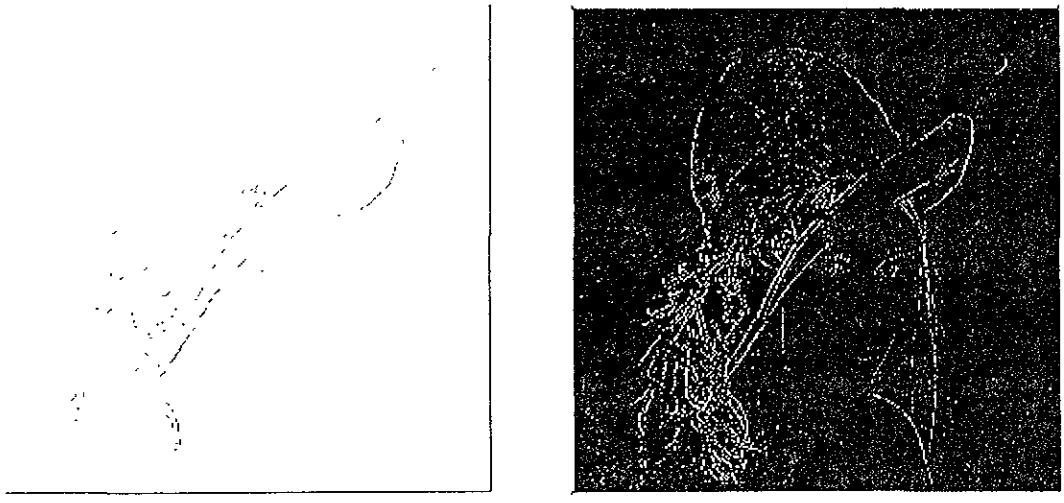
$$\nabla^2 h(x,y) = \left(\frac{r^2 - \sigma^2}{\sigma^4}\right) \exp\left(-\frac{r^2}{2\sigma^2}\right) \tag{3.12}$$

La figura 3.12.a) muestra una sección transversal de ésta función circularmente simétrica. Se observa su suavizado, los pasos por cero en $r = \pm\sigma$, el centro positivo y los extremos negativos. Esta forma es la base de la ecuación 3.10 y la máscara de la figura 3.11. Cuando se observa en una perspectiva tridimensional con el eje vertical como la intensidad, la ecuación 3.12 tiene la clásica forma de un *sombrero mexicano*. La figura 3.12.b) muestra tal representación, pero en forma de imagen. Se puede demostrar que el valor medio del operador laplaciano $\nabla^2 h$ es cero. Lo mismo es cierto para una imagen laplaciana obtenida por convolución de este operador con una imagen dada.

La figura 3.13 muestra un ejemplo en donde se aplican los conceptos descritos.

a) b)

Figura 3.13. a). Imagen original. **b).** Resultado de aplicar el laplaciano.



c) d)

Figura 3.13. c). Resultado invertido. **d).** Resultado umbralizado.

3.3.4. Enlazado de bordes

Si los bordes están marcados fuertemente y el nivel de ruido es bajo, se puede umbralizar una imagen de bordes y adelgazar la imagen binaria resultante hasta el nivel pixel, con el objeto de conseguir fronteras conectadas. En condiciones poco menores a las ideales, tal imagen de fronteras tendrá huecos que deberán ser llenados.

Los huecos pequeños pueden ser llenados simplemente al buscar en un vecindario reducido, por ejemplo 5X5; centrado el vecindario en un punto final, posteriormente para otros puntos finales y entonces llenar con pixeles frontera hasta conectar las fronteras en inspección. Desgraciadamente, en escenas complejas con muchos puntos finales, se puede sobresegmentar la imagen. Para impedir la sobresegmentación, se requiere que los dos puntos finales estén en concordancia en cuanto a distancia y orientación, dentro de una tolerancia predeterminada, antes de ser conectados.

Búsqueda heurística

Suponga que tenemos lo que aparenta ser un hueco en una frontera sobre una imagen de bordes, pero es demasiado largo para llenarlo adecuadamente mediante una línea recta o realmente no es un hueco o ambas cosas. Se puede establecer, como una medida de calidad, una función que pueda ser evaluada para cada ruta de conexión entre dos puntos finales, a los cuales llamamos *A* y *B*. Esta *función de calidad de bordes* puede incluir el promedio de un tipo de *fuerza* de los puntos y quizás alguna medida de su orientación angular.

Se empieza al evaluar los vecinos de *A* como candidatos a tomar el primer paso hacia *B*. Normalmente sólo los tres vecinos de A que yacen en la dirección general de *B* pueden ser considerados. Se selecciona el que

maximiza la función de calidad de borde de A al punto en cuestión. Entonces éste último se convierte en el punto de inicio para la siguiente iteración. Cuando finalmente se alcanza el punto B, la función de calidad de borde es comparada contra un umbral. Si el nuevo borde creado no es suficientemente fuerte es descartado.

Las técnicas de búsqueda heurística son computacionalmente caras si la función de calidad de borde es compleja y los huecos son grandes y abundantes. Tales técnicas tienen buen desempeño en imágenes simples, pero no necesariamente convergen sobre la ruta optima entre puntos.

Ajuste a curva

Si los puntos borde están generalmente esparcidos, puede ser deseable ajustar a un segmento lineal o a una curva spline a través de ellos para establecer una frontera adecuada y poder extraer los objetos. Existe un gran número de técnicas para el ajuste de curvas. Aquí sólo mencionaremos un método de ajuste utilizando segmentos lineales llamado *ajuste iterativo de puntos finales*.

Suponga que tenemos un grupo de puntos borde esparcidos entre dos puntos particulares A y B, y deseamos seleccionar el subconjunto de ellos para formar los nodos de una ruta con segmento lineal entre A y B. Empezamos por establecer una línea recta entre A y B. Entonces calculamos la distancia perpendicular desde la línea a los puntos restantes. El más alejado se convierte en el siguiente nodo en la ruta, que ahora tiene dos brincos. El proceso se repite para cada nuevo brinco de la ruta, hasta que no existan puntos borde remanentes mas alejados que una distancia predeterminada. Cuando esto se ha realizado para pares de puntos (A, B) a lo largo del objeto, se produce una aproximación poligonal de la frontera real.

Transformada de Hough

La línea recta $y = m\ x + b$ se puede representar en coordenadas polares como

$$\rho = x \cos(\theta) + y \operatorname{sen}(\theta) \tag{3.13}$$

en donde (ρ, θ) definen un vector del origen al punto más cercano de la línea, figura 3.14.a).

Podemos considerar un espacio de dos dimensiones, definido por los dos parámetros ρ y θ. Cualquier línea en el plano x, y representa a un punto en ese espacio. Entonces, la transformada Hough efectivamente transforma una línea recta en el espacio x, y a un punto en el espacio ρ, θ.

Ahora consideremos un punto particular (x_1, y_1) en el plano x, y. Existen un gran número de líneas que pasan por éste punto, y cada una de las líneas representan a un punto en el espacio ρ, θ. Estos puntos, sin embargo, cumplen con la ecuación 3.13, con x_1 y y_1 como constantes. Entonces el lugar de todas las líneas en el espacio x, y es una senoide en el espacio de los parámetros, y

cualquier punto en el plano x, y (figura 3.14.b)) corresponde a una senoide en el espacio ρ, θ (figura 3.14.c)).

Si tenemos un conjunto de puntos x_i, y_i que están en una línea recta con parámetros ρo y θo, entonces cada punto borde representa a una curva en el espacio ρ, θ. Sin embargo, todas las curvas cruzan el punto (ρo, θo), ya que es la línea que tienen en común (figura 3.14.c)).

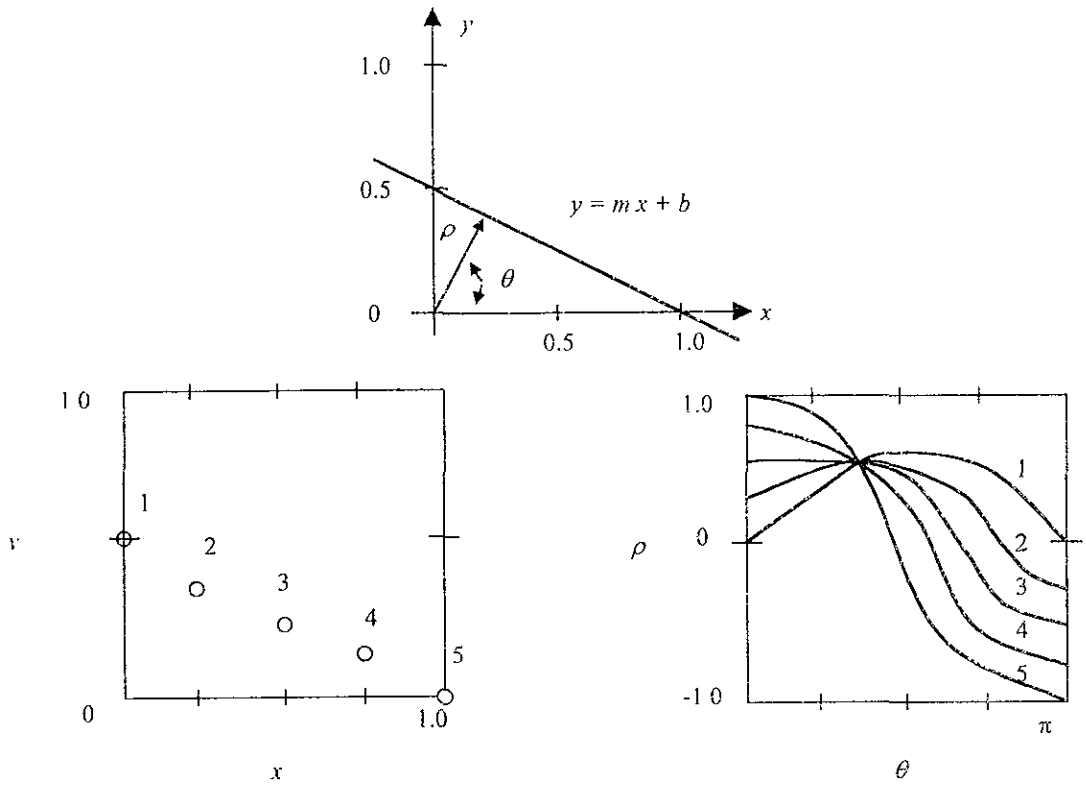


Figura 3.14. Transformada de Hough. **a).** Expresión en coordenadas polares para una línea recta. **b).** Plano x, y. **c).** Plano ρ, θ.

Para encontrar el segmento de línea recta al cual se ajusten los puntos, podemos especificar un histograma bidimensional en el espacio ρ, θ. Para cada punto borde, (x_i, y_i), incrementamos los pares del histograma en el espacio ρ, θ que correspondan a la transformada Hough (curva senoidal) para ese punto. Cuando se haya realizado para todos los puntos borde, el par conteniendo (ρo, θo) será un máximo local. Entonces, buscamos el histograma de espacio ρ, θ para un máximo local y obtener los parámetros de los segmentos lineales en la frontera.

3.4. Segmentación basada en regiones

El objetivo de la segmentación es dividir una imagen en regiones. En la sección 3.3 se ha planteado éste problema encontrando límites entre regiones basándose en discontinuidades de la intensidad, mientras que en la sección 3.2 se logró la segmentación por medio de umbrales basándose en las propiedades de distribución de pixeles, tales como la intensidad. En ésta sección se presentan algunas técnicas de segmentación que están basadas en encontrar directamente las regiones.

3.4.1. Crecimiento de regiones

Como su nombre implica, el crecimiento de regiones es un procedimiento que agrupa pixeles o subregiones dentro de regiones más grandes. La sencillez de éste método radica en la agregación de pixeles, que comienza con un conjunto de puntos generadores a partir de los que van creciendo las regiones al agregar a cada uno de estos puntos los pixeles vecinos que tienen propiedades similares. Para ilustrar éste procedimiento consideremos la figura 3.15.a), en la que los números dentro de las células representan valores de gris. Vamos a utilizar como generadores los puntos de coordenadas (3, 2) y (3, 4). Empleando dos puntos de salida se obtiene una segmentación que consiste en, al menos, dos regiones: R_1 asociada con el generador (3, 2), y R_2 asociada con el generador (3, 4). La propiedad P que se utiliza para incluir un pixel en una de las dos regiones es que la diferencia absoluta entre el nivel de gris de ese pixel y el generador sea menor que un umbral T. Cualquier pixel que satisfaga esa propiedad simultáneamente para ambos generadores se asigna a la región R_1. La figura 3.15.b) muestra el resultado que se obtiene al utilizar T=3. En éste caso la segmentación consiste en dos regiones, en las que los puntos de R_1 están señalados como a y los de R_2 como b. Se observa que cualquier punto de salida de cualquiera de estas dos regiones resultantes podría proporcionar el mismo resultado. Sin embargo, eligiendo T=8, se podría obtener una región única, como muestra la figura 3.15.c).

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

a)

a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b

b)

a	a	a	a	a
a	a	a	a	a
a	a	a	a	a
a	a	a	a	a
a	a	a	a	a

c)

Figura 3.15. Ejemplo de crecimiento de regiones. **a).** Matriz imagen original. b). Resultado de la segmentación al utilizar T=3. **c).** Resultado de la segmentación al utilizar T=8.

La ilustración anterior, aunque sencilla, muestra algunas de las dificultades fundamentales del crecimiento de regiones. Los dos problemas inmediatos son la selección de los generadores iniciales que representen correctamente a las regiones de interés y la selección de las propiedades adecuadas para la inclusión de puntos en las diversas regiones durante el proceso de crecimiento. La selección de uno o más puntos de salida normalmente puede basarse en la naturaleza del problema. Por ejemplo, en las aplicaciones militares de imágenes infrarrojas, los objetivos de interés están generalmente más calientes (y por ello aparecen más brillantes) que el fondo. La elección de los pixeles de más brillo es, por lo tanto, el punto de salida natural para un algoritmo de crecimiento de regiones. Cuando no se dispone de información a priori, el procedimiento consiste en calcular para cada pixel el mismo conjunto de propiedades que se utilizarán al final para asignar pixeles a las regiones durante el proceso de crecimiento. Si el resultado de estos cálculos muestra agrupaciones de valores, los pixeles cuyas propiedades los sitúan cerca del centro de estas agrupaciones se pueden utilizar como generadores. Por ejemplo, en la ilustración anterior, un histograma de nivel de gris podría mostrar que los puntos con intensidad 1 y 7 son los más predominantes.

La selección de criterios de similitud depende no solamente del problema, sino también del tipo de datos de imagen de los cuales se dispone. Por ejemplo, el análisis de las imágenes de satélite para trabajos terrestres depende mucho del empleo del color. Este problema puede ser significativamente más difícil de tratar cuando sólo se utilizan imágenes monocromas. Por desgracia, el disponer de imágenes multiespectrales o de algún tipo complementario es la excepción de la regla en el procesamiento de imágenes. Normalmente el análisis de la región debe realizarse con un conjunto de descriptores basados en la intensidad y en las propiedades espaciales (tales como momentos o texturas) de una imagen fuente única.

Los descriptores pueden proporcionar resultados falsos si no se utiliza la información de conectividad o adyacencia en el proceso de crecimiento de regiones. Por ejemplo, visualizar una distribución aleatoria de pixeles con

sólo tres valores de intensidad distintos. El reagrupar pixeles con la misma intensidad para formar una región sin prestar atención a la conectividad puede generar un resultado de segmentación que no sea congruente.

Otro problema del crecimiento de regiones es la formulación de una regla de parada. Básicamente, el crecimiento de una región debe detenerse cuando no hay más pixeles que satisfagan el criterio para su inclusión en la región. Hemos indicado que criterios como la intensidad, textura y color son locales por naturaleza y no tienen en cuenta la historia del crecimiento de la región. Otros criterios adicionales que incrementan la potencia del algoritmo de crecimiento de regiones utilizan el concepto de tamaño, de la semejanza entre un pixel candidato y los pixeles del crecimiento anterior (como por ejemplo, la comparación entre la intensidad de un candidato y la intensidad media de la región), y de la forma de la región que está en fase de crecimiento. El empleo de este tipo de descriptores está basado en la suposición de que se pueda disponer por lo menos de un modelo de resultados esperados.

Otro tipo de algoritmos definen una clase de función de energía asociada a la segmentación. Tal función mide simultáneamente la calidad en la suavidad de la curva frontera, la homogeneidad de la región e impide la segmentación excesiva.

La figura 3.16 muestra un ejemplo de crecimiento de regiones sobre una imagen en escala de gris.

a) b)

Figura 3.16. Crecimiento de regiones en escala de gris. **a).** Punto generador. **b).** Región después del crecimiento.

3.4.2. División de la imagen

Durante éste paso, la imagen se divide recursivamente en regiones pequeñas de acuerdo a una estructura del tipo *árbol cuádruple*. Esto significa que cada región es dividida simétricamente en cuatro o dejada intacta. La decisión para dividir la imagen está basada en un criterio de homogeneidad, de forma que la división sigue estrictamente el patrón de variación en la intensidad de la imagen.

En ésta sección primero presentamos el criterio de homogeneidad utilizado, posteriormente el algoritmo del árbol cuádruple y un ejemplo de división.

Criterio de homogeneidad

El criterio de homogeneidad puede aplicarse por medio de un predicado lógico que opere sobre todos los puntos de la región R_i. Para el criterio de homogeneidad mediante el error cuadrático medio (EQM), el predicado lógico P_{EQM} aplicado sobre la región R_i, es decir $P_{EQM}(R_i)$, puede tomar solamente dos valores posibles

$$P_{EQM}(R_i) = \begin{cases} VERDADERO \text{ si } EQM(R_i) \leq u_{EQM} \\ FALSO \text{ si } EQM(R_i) > u_{EQM} \end{cases} \tag{3.14}$$

en donde

$$EQM(R_i) = \sqrt{\sum^{N} \frac{\mu_{R_i} - R_i(x,y)}{N}}$$

N es el tamaño de la región R_i

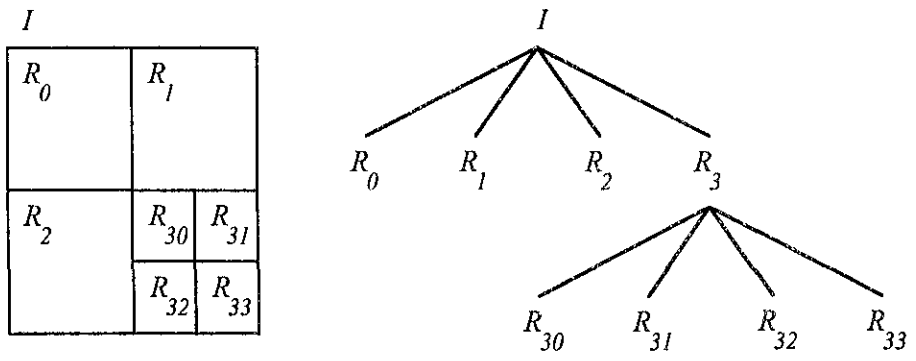
$\mu_{R_i} = \sum^{N} \frac{R_i(x,y)}{N}$ es la media aritmética

u_{EQM} es un umbral de comparación.

En resumen, si $P_{EQM}(R_i) = VERDADERO$ la región R_i es homogénea. Por el contrario, si $P_{EQM}(R_i) = FALSO$ la región R_i no es homogénea.

Algoritmo del árbol cuádruple

Para una imagen I el método de segmentar por árbol cuádruple consiste en subdividirla sucesivamente en cuadrantes cada vez más pequeños de forma que, para cualquier región disjunta R_i, de I, $P_{EQM}(R_i) = VERDADERO$. Esto es, si $P_{EQM}(R_i) = FALSO$, se divide la región en cuadrantes. Si P_{EQM} es $FALSO$ para cualquier cuadrante obtenido, se subdivide el cuadrante en subcuadrantes y así sucesivamente hasta un tamaño de región mínima determinado. El proceso efectivamente encuentra una representación gráfica en forma de árbol como se muestra en la figura 3.17.



a) b)

Figura 3.17. a) Imagen dividida. **b)** árbol cuádruple correspondiente.

El resultado de la división cumple con las siguientes características necesarias para un proceso de segmentación por regiones

$$\bigcup_{i=0}^{n} R_i = I$$

R_i es una región conexa, $i = 0, 1, 2, \ldots, n$,

$R_i \cap R_j = \varnothing$ para todo i y j $i \neq j$,

$P(R_i) = VERDADERO$ para $i = 0, 1, 2, \ldots, n$,

$P(R_i \cup R_j) = FALSO$ para $i \neq j$ (3.15)

Ejemplo de división por árbol cuádruple

En la figura 3.18 mostramos un ejemplo de división por árbol cuádruple sobre la imagen *Lena* después del filtro por reconstrucción 256X256X256, para un umbral $u_{EQM} = 12$, tamaño de región mínima igual a 1 pixel, resultando un total de 10432 regiones que cumplen con las condiciones en el conjunto de ecuaciones 3.14 y el criterio de homogeneidad en 3.13.

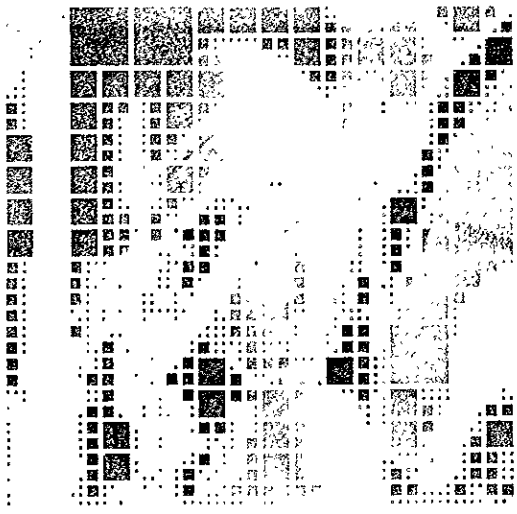


Figura 3.18. *Lena* después del filtro por reconstrucción dividida por árbol cuádruple, u_{EQM} = 12, 1 pixel tamaño de región mínima, 10432 regiones.

3.4.3. Fusión de regiones

La etapa de división produce una imagen sobresegmentada. Si bien sigue el patrón básico de intensidades, ésta tiende a crear particiones adyacentes con propiedades similares. Por ejemplo, una imagen siempre es dividida en cuatro aunque no sea necesario; podrían ser suficientes dos o tres divisiones. El inconveniente puede remediarse al fusionar regiones adyacentes similares.

Una manera eficiente de fusionar se asemeja bastante a la concepción visual que el observador se forma de una imagen; las personas observan la imagen y mentalmente realizan una clase de segmentación aleatoria, en donde distinguen objetos o regiones y los separan de la escena. Nuestro algoritmo intenta trabajar en una forma similar, en la medida de lo posible, al seleccionar regiones pivote aleatoriamente y examinar si sus vecinos son similares para fusionarlos. El proceso trabajando aleatoriamente, puede fusionar eventualmente regiones que en un primer intento no lo fueron. El desempeño anterior es posible ya que el parámetro que describe a regiones pequeñas aisladas tiende a adaptarse a una parámetro regional, conforme otras regiones pequeñas adyacentes van fusionándose.

La división y fusión de regiones reduce el esfuerzo computacional en comparación con la técnica de segmentación conocida como *crecimiento de regiones y fusión*. La razón es que la división y fusión opera sobre regiones en lugar de pixeles. Por lo tanto la técnica de segmentación presentada aquí se plantea como una opción realizable en cuanto a tiempo de procesamiento.

En ésta sección primero presentamos un criterio de similitud para la fusión, congruente con el proceso de división, al considerar un buen parámetro descriptor para cada región dividida, y fusionar sobre la base de la similitud de descriptores. Posteriormente planteamos el algoritmo para la fusión,

enfatizando en el progreso que exponemos como parte de éste trabajo, relacionado a la capacidad de fusión. A continuación presentamos un guía como ejemplo de implementación orientado a la programación en lenguaje C y finalmente un ejemplo práctico sobre la imagen estándar *Lena*.

Criterio de similitud

La división de acuerdo al criterio de homogeneidad basado en el EQM asegura, en cierta medida, que dos regiones adyacentes tienen la menor desviación sobre su media aritmética. Es decir, una región después de la división se puede considerar aproximadamente homogénea, con parámetro representativo y descriptivo a la media. De ésta manera, considerando a la media como parámetro fuertemente representativo de las regiones después de la división, es congruente considerar el proceso de fusión de acuerdo al criterio de similitud basado en la media.

El criterio de similitud entre dos regiones basado en la media (MED) puede aplicarse por medio de un predicado lógico P_{MED} que opere sobre todos los puntos de dos regiones R_i y R_j. Es decir $P_{MED}(R_i, R_j)$ puede tomar solamente dos valores posibles:

$$P_{MED}(R_i, R_j) = \begin{cases} VERDADERO\ si\ abs(\mu_{Ri} - \mu_{Rj}) \leq \mu_{MED} \\ FALSO\ si\ abs(\mu_{Ri} - \mu_{Rj}) > \mu_{MED} \end{cases} \qquad (3.16)$$

en donde

$\mu_{Ri} = \sum^{N} \frac{R_i(x,y)}{N}$ es la media aritmética

N es el tamaño de la región R_i

μ_{MED} es un umbral de comparación.

En resumen, si $P_{MED}(R_i, R_j) = VERDADERO$ las regiones R_i y R_j son similares. Por el contrario, $P_{MED}(R_i, R_j) = FALSO$ las regiones R_i y R_j no son similares.

Descripción del algoritmo

El algoritmo trabaja con base a la selección de una región pivote R_i en forma aleatoria. Entonces se examinan los vecinos R_{ij} mediante el enunciado lógico para la fusión:

si $P_{MED}(R_i, R_{ij}) = VERDADERO$ entonces R_i, R_{ij} se fusionan.

si $P_{MED}(R_i, R_{ij}) = FALSO$ entonces R_i, R_{ij} no se fusionan.

Para la siguiente discusión refiérase a la figura 3.19.

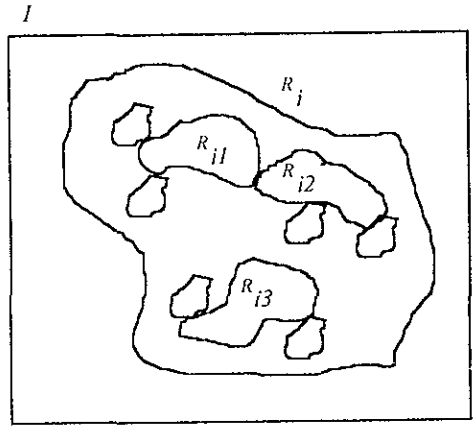


Figura 3.19. Descripción del algoritmo para la fusión.

Asumamos que R_i es una región con media μ_{Ri} de la imagen I, la cual es planteada como un posible resultado de la fusión. Las regiones R_{i1}, R_{i2}, y R_{i3} son regiones resultado del proceso de división, que en forma inherente, tienen la capacidad de fusionarse mayormente, y por lo tanto, la probabilidad de ser seleccionadas como regiones pivote crece al aumentar su tamaño. Supongamos también que en el inicio del algoritmo $P_{MED}(R_{i1}, R_{i2}) = FALSO$, es decir, R_{i1} y R_{i2} no son fusionables. Sin embargo, al avanzar el proceso $\mu_{Ri1} \rightarrow \mu_{Ri}$ y $\mu_{Ri2} \rightarrow \mu_{Ri}$ ya que otras regiones adyacentes pequeñas se fusionan a R_{i1} y R_{i2} en forma aislada. El algoritmo trabaja por tres aspectos principales: Las regiones pueden seleccionarse como pivotes en más de una ocasión, la probabilidad de seleccionar regiones grandes como pivotes crece al aumentar su tamaño. y al fusionar regiones, el parámetro descriptor de la región pivote (media) se adapta progresivamente al parámetro de la región global.

La discusión anterior se limitó a explicar el fenómeno entre regiones conexas como R_{i1} y R_{i2}. No obstante es sencillo pronosticar que un fenómeno similar sucede entre regiones no conexas, como por ejemplo entre R_{i1} y R_{i3} de la figura 3.19.

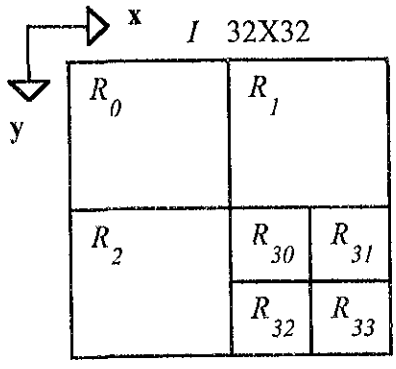
Guía para la implementación en lenguaje C

El algoritmo anterior puede fácilmente transportarse al lenguaje de programación C, considerando las siguientes estructuras de datos:

```
typedef struct {
  int Divi;       /* # de División */
  int X0;         /* Coordenada X0 */
  int Y0;         /* Coordenada Y0 */
  int Tam;        /* Tamaño */
  int Reg;        /* Región a la que
                  /* pertenece
} DIVISION;

typedef struct {
  int Reg;
} VECREG;
```

DIVISION se utiliza como un arreglo para almacenar el resultado de la división por árbol cuádruple. La figura 3.20 ilustra los parámetros de la estructura DIVISION en forma gráfica sobre una imagen de 32X32 después de la división.



a)

Num.	Divi.	X0	Y0	Tam.	Reg.
R0	0	0	0	16X16	0
R1	1	16	0	16X16	1
R2	2	0	16	16X16	2
R30	3	16	16	8X8	3
R31	4	24	16	8X8	4
R32	5	16	24	8X8	5
R33	6	24	24	8X8	6

b)

Figura 3.20. Descripción de la estructura DIVISION. a) árbol cuádruple. b) datos correspondientes después de la división y antes de la fusión.

Después de la división y antes de la fusión el parámetro REG (Región) es igual a DIVI (División), ya que la división aún no se ha asignado a ninguna región. Conforme el algoritmo avance, únicamente el parámetro REG se verá afectado.

VECREG se utiliza como un arreglo para almacenar temporalmente el índice de regiones vecinas.

Entonces con las siguientes definiciones de variables

```
DIVISION *division;
DIVISION *DivEnReg;
DIVISION *DivEnRegVec;
VECREG *RegVec;
```

y después de su correspondiente alojamiento en memoria, el algoritmo puede implementarse de la siguiente forma

```
Seleccionar un número de región aleatoria
Buscar en división[1...rango] el número de región y almacenar en
DivEnReg[1...NDiv] las divisiones que pertenecen a la región
Calcular la media de las divisiones en DivEnReg[1...NDiv] y almacenar
en MED1
PARA l=1 HASTA Ndiv
  Buscar regiones vecinas a partir de
  DivEnReg[1...NDiv] y almacenar en
  RegVec[1...cont]
FIN PARA
PARA l=1 HASTA cont
Buscar en división[1...rango] la región
  vecina en RegVec[l] y almacenar en
  DivEnRegVec[1...NDiv2] las divisiones que
  pertenecen a la región
  Calcular la media de las divisiones en
```

```
    DivEnRegVec[1...NDiv2] y almacenar en MED2
    Si abs(MED1-MED2)≤umbral entonces las
    divisiones en DivEnRegVec[1...NDiv2]
    pertenecen a la región.
FIN PARA
```

Algunas mejoras adicionales pueden implementarse. Por ejemplo se debe tener en cuenta que regiones de tamaño pequeño no pueden absorber regiones grandes. También el algoritmo puede finalizar cuando un cierto porcentaje de las regiones se ha cubierto.

Ejemplo de fusión de regiones

La figura 3.21 muestra a *Lena* después del filtrado, división y fusión de regiones.

Figura 3.21. *Lena* 256X256X256 después del filtrado, división y fusión. u_{EQM}=12, μ_{MED} =12.

Podemos observar que existe un cierto número de regiones pequeñas, agrupadas principalmente en los bordes y zonas de alto contraste. La fusión sigue siendo una imagen sobresegmentada aunque cumple con las condiciones en la ecuación 3.15.

3.4.4. Eliminación de regiones pequeñas y control de regiones

Muchas regiones de menor relevancia perceptualmente, permanecen después del procesamiento hasta ahora descrito. Tales regiones son usualmente muy contrastantes con sus vecinas, y por lo tanto no pueden ser fusionadas en grandes regiones más relevantes perceptualmente. Estas pequeñas regiones no son congruentes con una segmentación adecuada, usualmente proporcionan resultados desagradables para la visión humana. Lo adecuado es que dichas regiones sean absorbidas por el fondo o por grandes regiones adyacentes.

En ésta sección presentamos un criterio y algoritmos muy sencillos para eliminar regiones pequeñas, acompañados por un ejemplo práctico. Posteriormente desarrollamos el control en el número de regiones y un ejemplo práctico.

Eliminación de regiones pequeñas

En la práctica la eliminación se hace siempre al fusionar las pequeñas regiones hacia las regiones similares adyacentes, o hacia la región adyacente lidereando la mayor homogeneidad.

Criterio de tamaño

El criterio para la descripción de regiones basado en el tamaño relativo a la imagen total (TAM) puede aplicarse por medio de un predicado lógico P_{TAM} que opere sobre todos los puntos de la región R_i, en donde $R_i \subset I$. Es decir $P_{TAM}(R_i, I)$ puede tomar solamente dos valores posibles:

$$P_{TAM}(R_i, I) = \begin{cases} VERDADERO\ si\ \dfrac{N}{M} \leq \tau_{TAM} \\ FALSO\ si\ si\ \dfrac{N}{M} > \tau_{TAM} \end{cases} \tag{3.17}$$

en donde

N es el tamaño de la región R_i

M es el tamaño de la imagen I

τ_{TAM} es un umbral de comparación.

En resumen, si $P_{TAM}(R_i, I) = VERDADERO$ la región R_i es pequeña. Por el contrario, $P_{TAM}(R_i, I) = FALSO$ la región R_i no es pequeña.

Descripción del algoritmo

El algoritmo trabaja con base a la fusión de regiones pequeñas con su vecino lidereando la mayor similitud. Asumamos que $P_{TAM}(R_i, I) = VERDADERO$. Entonces se examinan todos los vecinos R_{ij} con el objeto de analizar el vecino con la mayor similitud, sobre la base del mejor descriptor (media aritmética), es decir:

$\min(\text{abs}(\mu_{Ri} - \mu_{Rij}))$

Ejemplo de eliminación de pequeñas regiones

La figura 3.22 muestra a *Lena* después de la eliminación de regiones pequeñas, $\tau_{TAM} = 0.012\%$.

Figura 3.22. *Lena* 256X256X256 después del filtrado división, fusión y eliminación de regiones pequeñas. $u_{EQM} = 12$, $\mu_{MED} = 12$, $\tau_{TAM} = 0.012\%$.

Control del numero de regiones

Este es el paso final en la segmentación espacial de imágenes. El objetivo principal es controlar el resultado de la segmentación en términos de un número final de regiones.

Descripción del proceso

La técnica utilizada es simplemente similar a la fusión en el sentido de que las regiones son fusionadas, pero ahora el proceso finaliza cuando se alcanza el número de regiones deseado. Ya que ésta etapa produce sucesivamente imágenes segmentadas con un número decreciente de regiones, ésta se puede ver como un generador jerárquico con niveles de simplificación decrecientes.

Ejemplo del control en el número de regiones

La figura 3.23 muestra el resultado de controlar el número de regiones finales en la segmentación espacial.

Figura 3.23. *Lena* 256X256X256 después del filtrado, división, fusión, eliminación de regiones pequeñas y control del número de regiones. u_{EQM}=12, μ_{MED}=12, τ_{TAM} =0.012%, regiones finales = 68.

Capitulo 4

Segmentación espacio temporal

4.1. Introducción

En forma natural, la visión del ser humano acerca del mundo real cambia constantemente debido al movimiento alrededor. Aún cuando uno permanezca estático. los objetos se mueven, proporcionando una cantidad de información abundante que los seres humanos asimilamos intuitivamente. Inclusive para algunos animales, tal habilidad para analizar y comprender el movimiento alrededor, es esencial para la sobrevivencia; sin ella no podríamos dar continuidad a nuestras percepciones o anticipar eventos inminentes. En éste proceso de análisis intervienen procesos fisiológicos muy complejos que son transparentes y naturales en la vida diaria. Lo que para los seres humanos y algunos animales parece una labor cotidiana, para las máquinas no es una tarea trivial.

Restringiendo nuestro estudio de escenas en movimiento al proyectado en cámaras monoculares, lo que se requiere es una representación flexible del movimiento visual que pueda utilizarse para fines de análisis y cálculo mediante medios computacionales. En éste sentido, los trabajos de ésta área de estudio consideran el *flujo óptico* como una representación del movimiento real aparente proyectado en el plano imagen monocular. Estrictamente, a la proyección de cada punto ambiental visible se le puede asociar un vector de velocidad; el campo de vectores de velocidad es el flujo óptico. El flujo óptico puede ser estimado a partir de lo observable: el patrón de brillo espacio temporal capturado por un sistema de visión (video). La estimación se realiza a través de la solución de un conjunto de *restricciones* derivadas de la *hipótesis* realizada sobre la base del patrón de brillo.

Como es bien conocido, ya que estamos resolviendo un problema de estimación, en realidad se obtiene una aproximación que dependerá de las restricciones planteadas y del método de solución. Utilizando términos apropiados en el planteamiento del algoritmo, se puede ver al flujo óptico como el intento de significado de un algoritmo para la estimación del flujo óptico

El flujo óptico es el campo de velocidad 2D, describiendo el movimiento aparente en la escena, que resulta del movimiento independiente de objetos desde el punto de vista del observador. Específicamente, la figura 4.1 ilustra como la rotación y traslación del punto **P** hacia **P'** en coordenadas globales ocasiona el movimiento del punto **p** hacia **p'** proyectado en coordenadas del plano imagen.

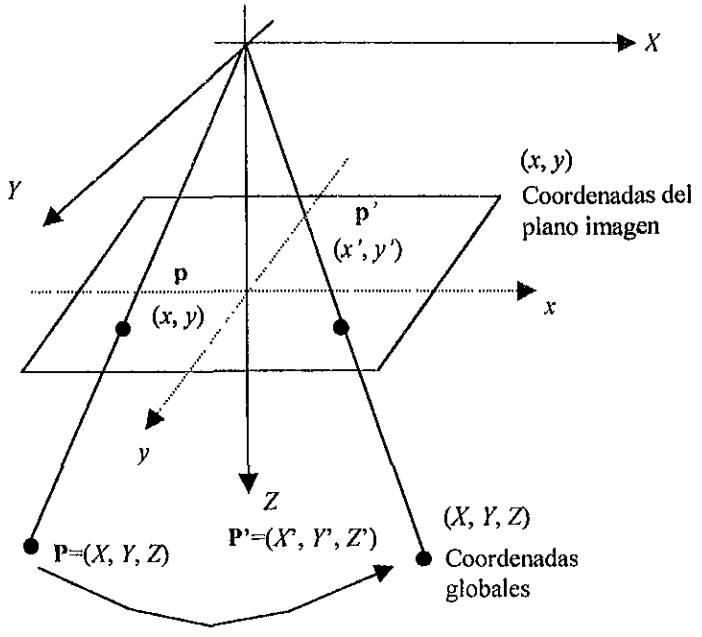


Figura 4.1. Movimiento 3D proyectado en el plano 2D.

El análisis del movimiento en imágenes puede desempeñar ciertas funciones importantes dentro del campo de visión por computadora:

1. *Detección por movimiento en la escena.* El movimiento en imágenes puede ser indicativo del movimiento real en la escena observada. En el caso de un sistema de visión estático, la detección del movimiento de la escena directamente proporciona la cantidad de movimiento en la imagen. En el caso de una cámara en movimiento, el fenómeno es más complejo e involucra etapas adicionales de procesamiento (por ejemplo, segmentación de movimiento e interpretación de regiones), o de compensación por movimiento en la cámara.
2. *Segmentación de objetos.* El movimiento en imágenes es un indicio para la segmentación, para la identificación de diferentes objetos en movimiento y para distinguir estos del fondo. La suposición es que los puntos dentro del contorno de oclusión de un objeto imagen tienen velocidades con variaciones suaves.
3. *Seguimiento.* El movimiento puede ser utilizado para la persecución de un objeto imagen sobre la base de su comportamiento dinámico actual.
4. *Medición de profundidad y movimiento ambiental.* La profundidad y movimiento relativos en el ambiente pueden ser relacionados

cuantitativamente con la posición y velocidad en la imagen. La información tridimensional infrarroja puede utilizarse para estimar forma en el espacio, y para determinar la posición y movimiento relativos del sistema de visión a los objetos en el ambiente. También puede considerarse una interpretación cualitativa.

Para realizar tales funciones en visión por computadora, los problemas formales se plantean mediante hipótesis acerca de la naturaleza de las imágenes registradas y del movimiento de regiones. En éste sentido los estudios para el análisis de movimiento en las imágenes son motivados por diversas aplicaciones del mundo real. Algunos de los dominios en donde el análisis de movimiento ha alcanzado un status especial son los siguientes:

1. *Televisión* para la codificación de video por compensación de movimiento, en varios servicios de telecomunicaciones como videoconferencia, televisión digital, y más recientemente, transmisión de señales de televisión con alta definición. El propósito del análisis de movimiento entonces es explotar la redundancia temporal para reducir la tasa de transmisión mientras se preserve la calidad de las imágenes reconstruidas recibidas.
2. *Robots móviles* (terrestres, acuáticos, espaciales). El propósito es conferir a los robots la capacidad de navegar autónomamente en ambientes parcialmente desconocidos (detección de obstáculos, posicionamiento, seguimiento de objetos en movimiento, etc.).
3. *Análisis de imágenes de satélite*, particularmente en meteorología para medir el movimiento de las nubes y establecer mapas de vientos.
4. *Aplicaciones militares* para el seguimiento de objetivos y navegación autónoma de diversos dispositivos tales como vehículos o proyectiles.

Algunos dominios más recientes de aplicación son los siguientes:

1. *Análisis de imágenes biomédicas.* Algunos ejemplos importantes son el análisis automático de los movimientos del corazón, o el estudio del movimiento humano (en medicina deportiva, re-educación, etc.)
2. *Vigilancia* de sitios (detección de intrusiones), y *monitoreo* de trafico urbano.
3. *Interfaces y realidad virtual.* Nuevas interfaces hombre máquina, especialmente en el naciente dominio de la realidad virtual, pueden ser requeridas en breve para interpretación de movimiento facial (por ejemplo movimiento de labios) o gestos humanos (por ejemplo movimiento de manos).

El presente capitulo está organizado como sigue: la segunda sección muestra un estudio de geometría clásica para el establecimiento del modelo de movimiento. La tercera sección presenta algunas restricciones de la visión monocular en la estimación del flujo óptico. La cuarta y quinta partes presentan algunas técnicas particulares para la estimación cuantitativa del

flujo óptico. La parte seis se complementa con el capítulo anterior y plantea los criterios para la segmentación espacio temporal así como un análisis del desempeño de la estimación de movimiento.

4.2. Modelo de movimiento

4.2.1. El modelo general para las variaciones temporales

Una imagen se forma por la proyección 3D de una escena en el plano imagen. Si la cámara captura dos imágenes sucesivas I_t e I_{t+1} en dos tiempos sucesivos t y $t+1$, la iluminancia en los pixeles puede cambiar. Estas modificaciones en la iluminancia vienen de diferentes fuentes (ver figura 4.2):

- Movimientos relativos entre la escena y la cámara.
- surgimiento u oclusión de algunas áreas.
- Variaciones de iluminación en la escena.
- Problemas de adquisición en el proceso de captura de imágenes.

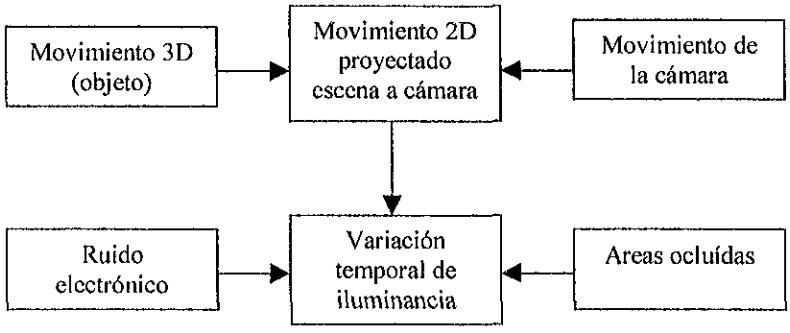


Figura 4.2. Variaciones temporales de iluminancia en imágenes.

Para el caso de un enfoque basado en regiones, es necesario analizar y estimar estas variaciones para tratar con tres problemas fundamentales: Modelado de las variaciones temporales, elección de un criterio de estimación y definición del estimador. Lo que resta del capitulo trata extensamente tales problemas, lo que resta de la sección lo discute brevemente

Modelado de las variaciones temporales

En el caso general, la predicción I'_{t+1} de la imagen real I_{t+1} a partir de la imagen I_t puede estimarse de acuerdo con la ecuación:

$$I'_{t+1}(\mathbf{p})=E[B(I_t, \mathbf{p}-\mathbf{d})+\gamma(\mathbf{p})] \qquad (4.1)$$

En donde B es un operador de interpolación, como el mostrado en la figura 1.3, usado para calcular la iluminancia de un punto posiblemente no entero ($\mathbf{p}$-

d). **d** representa el movimiento aparente 2D y $\gamma(\mathbf{p})$ la estimación en la variación de la iluminación en el punto **p**. Sea

$$I'_t(\mathbf{p}-\mathbf{d}) \equiv B(I_t, \mathbf{p}-\mathbf{d}) \tag{4.2}$$

y $E(x)$ definida como sigue

$$E(x) = \begin{cases} \text{ENT}\left(x+\frac{1}{2}\right), & \text{si } 0 \le \left(x+\frac{1}{2}\right) \le 255, \\ 0, & \text{si}\left(x+\frac{1}{2}\right) < 0 \\ 255, & \text{si}\left(x+\frac{1}{2}\right) > 255 \end{cases} \tag{4.3}$$

Note también que la función γ es diferente al error por compensación de movimiento. Esto es debido al hecho que γ está incluida en la función de interpolación.

Entonces si tenemos una estimación de movimiento **d** y una estimación en la variación de iluminación γ en el punto **p**, la imagen de movimiento compensado se calcula como

$$I_{t+1}(\mathbf{p}) = E[I'_t(\mathbf{p}-\mathbf{d}) + \gamma(\mathbf{p})] + \alpha(\mathbf{p}) \tag{4.4}$$

en donde $\alpha(\mathbf{p})$ es el error de compensación e iluminación en **p**. Existen varios modelos de iluminación y movimiento [13, 14, 25]; un problema entonces es elegir entre una de esas representaciones. Parte de nuestro trabajo es exponer algunas consideraciones a las representaciones mencionadas.

Movimiento y segmentación

La estimación se logra de acuerdo a un criterio de minimización Este criterio se calcula sobre la base de regiones, de manera que una modificación en la segmentación afecta el valor del criterio. Es bien conocido que las etapas de estimación y segmentación son altamente dependientes y en ocasiones se desarrollan como un proceso cooperativo [24]. Como consecuencia, la definición del movimiento 2D aparente depende de la región considerada. La figura 4.3 muestra una segmentación en nueve regiones disjuntas con movimiento aparente de traslación, pero la región entera se puede representar con movimiento de divergencia.

En las siguientes secciones, sólo tomamos en cuenta el modelo de movimiento basado en regiones.

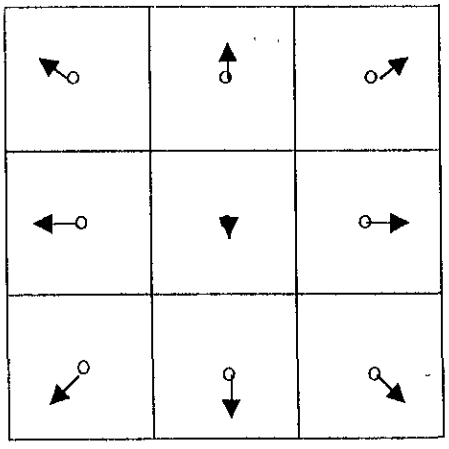


Figura 4.3. Influencia de la segmentación en la representación de movimiento.

4.2.2. Modelado

En la sección previa se propuso una expresión general para obtener la predicción de la imagen en un tiempo $t+1$ a partir de la imagen en t. En el contexto de codificación de secuencias de imágenes, se necesitan representaciones compactas en donde los descriptores de la variación temporal sean transmitidos (figura 1). Tal representación se puede obtener en forma sencilla utilizando aproximaciones polinomiales, o en el caso de movimiento, utilizando una proyección simplificada del movimiento 3D físico en el plano imagen, como se mostró en la figura 4.1. Las siguientes secciones presentan estos esquemas.

Antes, es conveniente resaltar la notación utilizada:

Θ representa un modelo de movimiento y $\Theta=(\theta_1 \ \ldots \ \theta_m)^t$ es el vector de parámetros de movimiento.

Γ Representa el modelo de variación de iluminación y $\Gamma=(\gamma_1 \ \ldots \ \gamma_n)^t$ su vector de parámetros.

$\Phi=(\Theta, \Gamma)$ representa el modelo de variación temporal y $\Phi=(\Theta, \Gamma)^t$ su estimación.

4.2.3. Modelo de movimiento

Para definir el modelo de movimiento, investigamos sucesivamente los dos enfoques usuales que consisten, la primera, en la introducción de restricciones matemáticas a priori (aproximación polinomial). En la segunda, el análisis de la proyección del movimiento 3D físico proporciona otros parámetros.

Descripción polinomial

En ésta sección asumimos que en una región dada j, el campo de vectores de movimiento pueden ser aproximados por una representación polinomial (la validez del modelo correspondiente puede verificarse a posteriori utilizando un criterio de evaluación). La expresión general de tal aproximación está dada por

$$\mathbf{d} = \sum_{n=0}^{k} \sum_{i=0}^{n} \left[(x - x_g)^{n-i} (y - y_g)^i \begin{pmatrix} a_{ni} \\ b_{ni} \end{pmatrix} \right] \tag{4.5}$$

en donde k es el orden de la aproximación polinomial, M_k el orden del k-esimo orden del modelo (vea la tabla 4.1) y $G(x_g, y_g)$ es el centro de gravedad 2D de una región (note que se usa como una referencia de región). Algunos modelos M_k son los siguientes

$$M_0 = \begin{pmatrix} a_{00} \\ b_{00} \end{pmatrix} \qquad M_1 = \begin{pmatrix} a_{10} & a_{11} \\ b_{10} & b_{11} \end{pmatrix} \qquad M_2 = \begin{pmatrix} a_{20} & a_{21} & a_{22} \\ b_{20} & b_{21} & b_{22} \end{pmatrix}$$

Proyección del movimiento físico 3D

En este esquema, se proyecta el movimiento físico 3D sobre el plano imagen. Para ello, se utiliza el torque cinemático (traslación y rotación sobre los ejes coordenados) 3D (T_x, T_y, T_z, θ_x, θ_y, θ_z) para calcular el desplazamiento 3D de un objeto rígido, ver figura 4.4.

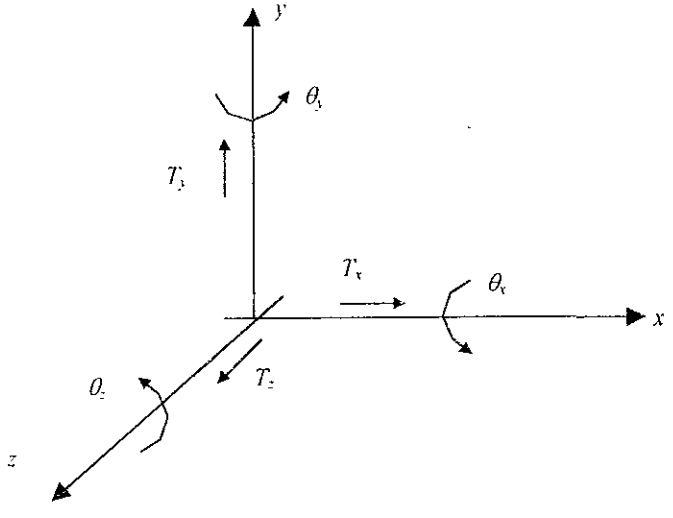


Figura 4.4. Torque cinemático 3D.

Las coordenadas 3D $(X_2, Y_2, Z_2)^t$ en el tiempo $t+1$ de un punto $P(X_1, Y_1, Z_1)$ se define como

$$\begin{pmatrix} X_2 - X_{g1} \\ Y_2 - Y_{g1} \\ Z_2 - Z_{g1} \end{pmatrix} = \begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix} + R_z R_y R_z \begin{pmatrix} X_1 - X_{g1} \\ Y_1 - Y_{g1} \\ Z_1 - Z_{g1} \end{pmatrix} \tag{4.6}$$

en donde $G_{g1}(X_{g1}, Y_{g1}, Z_{g1})^t$ es el centro de gravedad 3D de un objeto en el tiempo t y R_z R_y R_x la transformación de coordenadas

$$R_zR_yR_x = \begin{pmatrix} \cos\theta_z & -\operatorname{sen}\theta_z & 0 \\ \operatorname{sen}\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_y & 0 & -\operatorname{sen}\theta_y \\ 0 & 1 & 0 \\ \operatorname{sen}\theta_y & 0 & \cos\theta_y \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\operatorname{sen}\theta_x \\ 0 & \operatorname{sen}\theta_x & \cos\theta_x \end{pmatrix}$$

Usando tal modelo, es posible simplificar la ecuación de proyección sobre el plano imagen, ecuación 4.6, si se asume que:

1. Los objetos 3D son rígidos.
2. Se utiliza proyección en perspectiva.
3. No se toman en cuenta las rotaciones en el plano imagen ($\theta_x=\theta_y=0$).
4. En una región R, si Z_1 y Z_2 representan respectivamente los puntos más cercano y lejano hacia la cámara (a lo largo del eje Z), entonces se puede asumir que: $\frac{Z_2 - Z_1}{Z_1} << 1$. La suposición es valida si los objetos están lejos de la cámara o si la región considerada es pequeña.
5. La rotación de ángulos alrededor del eje óptico es pequeña ($\theta_z<<1$) (implicando que $\cos\theta_z=1$ y $\operatorname{sen}\theta_z=\theta_z$).

Bajo estas consideraciones (que pueden ser consideradas como hipótesis realistas para el caso de codificación de video), se calcula el desplazamiento **d** de un pixel **P** utilizando el vector de parámetros Θ_{MS} al utilizar

$$\mathbf{d}=M_0+M_a\ \mathbf{GP} \tag{4.7}$$

en donde

$$M_0 = \begin{pmatrix} t_x \\ t_y \end{pmatrix} = \begin{pmatrix} a_{00} \\ b_{00} \end{pmatrix} \text{ y } M_a = \begin{pmatrix} k & -\theta \\ \theta & k \end{pmatrix}$$

Entonces el parámetro de vectores 4D $\Theta_{MS}=(t_x, t_y, k, \theta)^t$ identifica los movimientos 3D (modelo lineal simplificado *MS*); los parámetros corresponden a las componentes 2D de las traslaciones aparentes (t_x, t_y), la relación de divergencia (k) y la rotación angular ($\theta=\theta_z$).

La eliminación de algunos parámetros lineales proporciona modelos aún más simplificados. Respectivamente, si eliminamos θ, k o ambos, se obtienen el *modelo de divergencia* (*MD*), el *modelo de rotación* (*MR*) y el *modelo constante* (*Mo*). Por el contrario, para obtener un modelo lineal completo (*ML*) (que es útil cuando la hipótesis previa no es realista) es necesario introducir otros parámetros lineales (parámetros hiperbólicos $h1$ y $h2$). Entonces se obtiene un nuevo vector de parámetros 6D $\Theta_{MS}=(t_x, t_y, k, \theta, h1, h2)^t$ [15]. Estos modelos se definen como se muestra a continuación (ver también la tabla 4.1).

Modelo de rotación

$$\mathbf{d}=M_0+M_r\,\mathbf{GP} \tag{4.8}$$

Modelo de divergencia

$$\mathbf{d}=M_0+M_d\,\mathbf{GP} \tag{4.9}$$

Modelo lineal

$$\mathbf{d}=M_0+(M_a+M_b)\mathbf{GP} \tag{4.10}$$

en donde

$$M_r=\begin{pmatrix}0 & -\theta\\ \theta & 0\end{pmatrix} \quad Md=\begin{pmatrix}k & 0\\ 0 & k\end{pmatrix} \quad M_b=\begin{pmatrix}h1 & h2\\ -h1 & h2\end{pmatrix}$$

Representación	Nombre del modelo	Ecuación
MN	Sin movimiento	$\Theta_{MN}=0$
M0 (k=0)	Constante	$\mathbf{d}=M_0$ $\Theta_{M0}=(a_{00}, b_{00})^t=(t_x, t_y)^t$
MR	Rotación	$\mathbf{d}=M_0+M_r\,\mathbf{GP}$ $\Theta_{MR}=(t_x, t_y, \theta)^t$
MD	Divergencia	$\mathbf{d}=M_0+M_d\,\mathbf{GP}$ $\Theta_{MD}=(t_x, t_y, k)^t$
MS	Lineal simplificado	$\mathbf{d}=M_0+M_a\,\mathbf{GP}$ $\Theta_{MS}=(t_x, t_y, k, \theta)^t$
ML	Lineal	$\mathbf{d}=M_0+(M_a+M_b)\,\mathbf{GP}$ $\Theta_{ML}=(t_x, t_y, k, \theta, h1, h2)^t$
M1 (k=1)	Afín	$\mathbf{d}=M_0+M_1\,\mathbf{GP}$ $\Theta_{M1}=(a_{00}, b_{00}, a_{10}, b_{10}, a_{11}, b_{11})^t$
M2 (k=2)	Cuadrático	$\mathbf{d}=M_0+M_1\,\mathbf{GP}+M_2[(x-x_g)^2, (x-x_g)(y-y_g), (y-y_g)^2]^t$ $\Theta_{M1}=(a_{00}, b_{00}, a_{10}, b_{10}, a_{11}, b_{11}, a_{20}, b_{20}, a_{21}, b_{21}, a_{22}, b_{22})^t$

Tabla 4.1. Ejemplos de modelos de movimiento y de variación de iluminación.

Γ_N	Sin iluminación	$\gamma(p)=0$
Γ_0 (k=0)	Orden cero	$\gamma(p)=\gamma_{00}$ $\Gamma_0=(\gamma_{00})$
Γ_1 (k=1)	Primer orden	$\gamma(p)=\gamma_{00}+\gamma_{10}(x-x_g)+\gamma_{11}(y-y_g)$ $\Gamma_1=(\gamma_{00}, \gamma_{10}, \gamma_{11})^t$

Tabla 4.1. Ejemplos de modelos de movimiento y de variación de iluminación (continuación).

4.2.4. Modelo de variación de la iluminación

En esta sección se propone y justifica una introducción a los modelos de variación en la iluminación (función γ en la ecuación 4.4). Para tal propósito, analizamos y definimos matemáticamente las variaciones de iluminación para un pixel, y después se extrapola para una región. Entonces introducimos la aproximación polinomial para dichas variaciones. Esta aproximación nos proporciona un conjunto de modelos de iluminación que se pueden incorporar a los algoritmos de estimación.

Representación basada en pixeles

Matemáticamente, si las variaciones temporales sólo provienen del movimiento y sí $\mathbf{d}_0$ representa la proyección en el plano imagen del desplazamiento 3D de un punto $\mathbf{p}$, entonces

$$I_{t+1}(\mathbf{p})-I_t(\mathbf{p}-\mathbf{d}_0)=0$$

En la práctica, tal suposición generalmente no es cierta, debido al ruido, variación de iluminación, áreas ocultas (en tales casos, el movimiento no tiene sentido físico). Por lo tanto, en general, tenemos

$$I_{t+1}(\mathbf{p})-I_t(\mathbf{p}-\mathbf{d}_0)=\alpha_0(\mathbf{p})$$

Se puede comprobar que existe una función $\gamma_0(\mathbf{p})$ que satisface

$$I_{t+1}(\mathbf{p})-E[I'_t(\mathbf{p}-\mathbf{d}_0)+\gamma_0(\mathbf{p})]=0$$

En donde $\gamma_0(\mathbf{p})$ representa la variación de iluminación 2D en el pixel p; note que debido a que la función E no es continua, $\gamma_0(\mathbf{p})$ no es única.

Considerando ahora una $\mathbf{d}_a$ que minimice $|DFD(\mathbf{p}, \mathbf{d})|$ ($\mathbf{d}_a$ no es única)

$$I_{t+1}(\mathbf{p})-I'_t(\mathbf{p}-\mathbf{d}_a)+\gamma_0(\mathbf{p})]=0$$

Existe una $\gamma_a(\mathbf{p})$ que satisface

$$I_{t+1}(\mathbf{p})-E[I'_t(\mathbf{p}-\mathbf{d}_a)+\gamma_a(\mathbf{p})]=0$$

Con $|\alpha_a(\mathbf{p})| \leq |\alpha_0(\mathbf{p})|$. $\gamma_a(\mathbf{p})$ representa la variación aparente de iluminación 2D en el pixel $\mathbf{p}$.

Representación basada en regiones

Debido a que se utilizan representaciones de modelos de movimiento prejuiciados, es difícil definir las variaciones físicas de iluminación. Efectivamente, no podemos separar el error residual, que se origina de las variaciones de iluminación, a partir de errores debidos a la perdida de concordancia entre el modelo y el movimiento físico. Como una consecuencia, existen diferentes esquemas para definir las variaciones de iluminación:

1. Utilización del movimiento físico 3D

   Esta solución consiste en la definición de variaciones 2D de la función de iluminación (Γ_0) usando la proyección del movimiento físico 3D. Entonces, los errores residuales son considerados como efectos de iluminación (y áreas ocluidas). Mientras la definición puede ser interesante en análisis de imágenes para su comprensión, en codificación no se reduce el error de reconstrucción; de manera que Γ_0 no es una estimación óptima.

2. Utilización del movimiento 2D aparente (Θ_a=arg $\min_\Theta$ $\{I_{t+1}(\mathbf{p})$-$E[I'_t(\mathbf{p}-\mathbf{d}_{\Theta a})]\}$)

$\Gamma_a(\mathbf{p})$ se define como sigue:

$$\Gamma_a(\mathbf{p}) = \arg min_\Gamma \sum_{\mathbf{p}\in j} (I_{t+1}(\mathbf{p}) - E[I'_t(\mathbf{p} - \mathbf{d}_{\Theta a}) + \gamma(\mathbf{p})])^2 \qquad (4.11)$$

3. Optimización global

   Este método consiste en una optimización global de los descriptores de movimiento e iluminación para obtener el movimiento aparente 2D (Θ_g) y los descriptores de iluminación (Γ_g). Entonces:

$$\Phi_g = (\Theta_g, \Gamma_g)^t = \arg min_{(\Theta,\Gamma)} \sum_{\mathbf{p}\in j} (I_{t+1}(\mathbf{p}) - E[I'_t(\mathbf{p} - \mathbf{d}_{\Theta a}) + \gamma(\mathbf{p})])^2 \qquad (4.12)$$

Para la aplicación en codificación de video en donde se desea minimizar el error de predicción, la mejor estimación consiste en adoptar el esquema de optimización global.

Aproximación polinomial

La manera más simple de modelar tal función de iluminación es realizar una aproximación polinomial. La aproximación limitada al orden k puede escribirse como:

$$\gamma(\mathbf{p}) = \sum_{n=0}^{k} \sum_{i=0}^{n} \gamma_{ni} \cdot (x - x_g)^{n-i} \cdot (y - y_g)^i \qquad (4.13)$$

4.3. Restricciones en el flujo óptico

Esta sección considera el problema de la recuperación del flujo óptico a partir de una secuencia de imágenes. Uno comienza por hacer algunas suposiciones acerca de la escena que son necesariamente idealizaciones y que son violadas comúnmente en la práctica. Tales suposiciones entonces son agrupadas en un conjunto de restricciones en la interpretación del movimiento en imágenes como se muestra en la figura 4.5. Esta sección explora el uso de las tres principales restricciones conocidas como: *conservación de los datos*, *coherencia espacial*, y *continuidad temporal*.

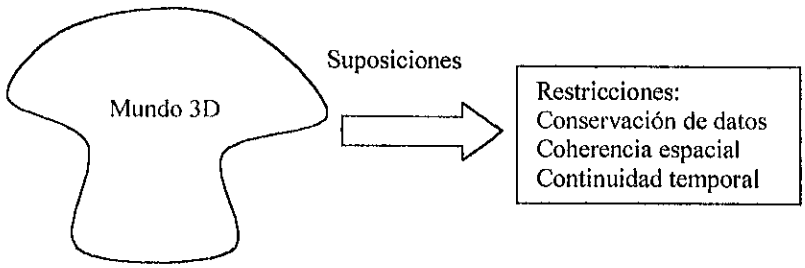


Figura 4.5. Restricciones en el flujo óptico.

4.3.1. Conservación de los datos

Los algoritmos para calcular el flujo óptico explotan comúnmente los cambios de intensidad en la imagen sobre el tiempo. Los enfoques más populares incluyen técnicas basadas en el gradiente [15], correlación [25], y filtrado espacio-temporal [21]. Los esquemas anteriores explotan la suposición de la conservación de los datos, también conocida como suposición de iluminación constante:

Las mediciones en la imagen (por ejemplo, intensidad en la imagen) correspondientes a una región pequeña permanecen constantes, sin embargo la localización de la región puede cambiar en el tiempo.

Lo anterior se muestra en la figura 4.6. La región marcada en la imagen 4.6.a) parece aproximadamente la región de la imagen 4.6.b), excepto que ésta se ha movido. Note que la región inclusive puede ocultar o mostrar regiones no observables previamente.

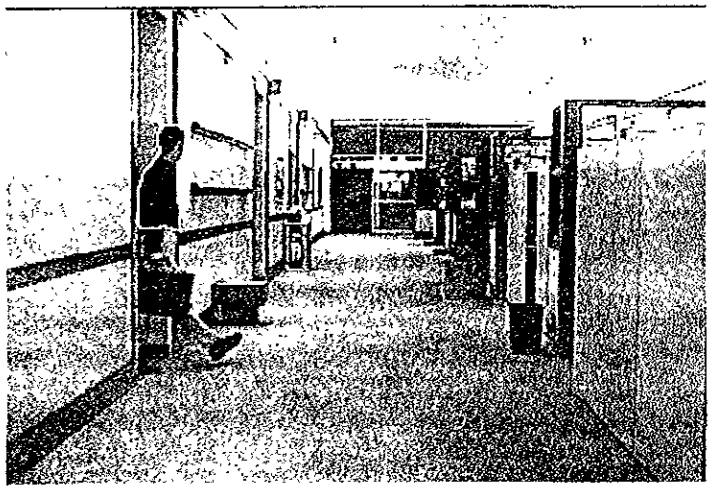


Figura 4.6.a) Restricciones en el flujo óptico.

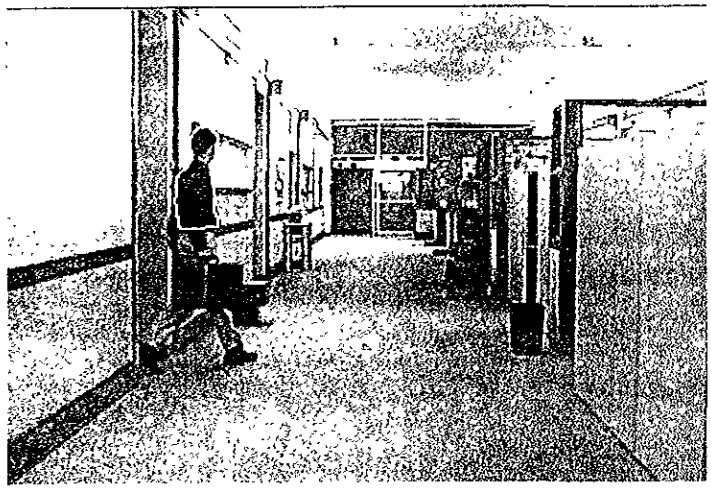


Figura 4.6.b) Restricciones en el flujo óptico.

4.3.2. Coherencia espacial

La sola restricción a la coherencia de los datos no siempre es suficiente para recuperar exactamente el flujo óptico. Primero, los estimadores de movimiento local, basados en la conservación de los datos, sólo restringen parcialmente la solución. Considere el movimiento de una línea como en la figura 4.7. Dentro de una pequeña región, la restricción en la conservación de los datos no puede determinar en forma única el movimiento de la línea; existe un gran número de interpretaciones que inclusive son consistentes con la restricción. Entonces cuando se visualiza a través de una pequeña apertura, el movimiento resulta ambiguo.

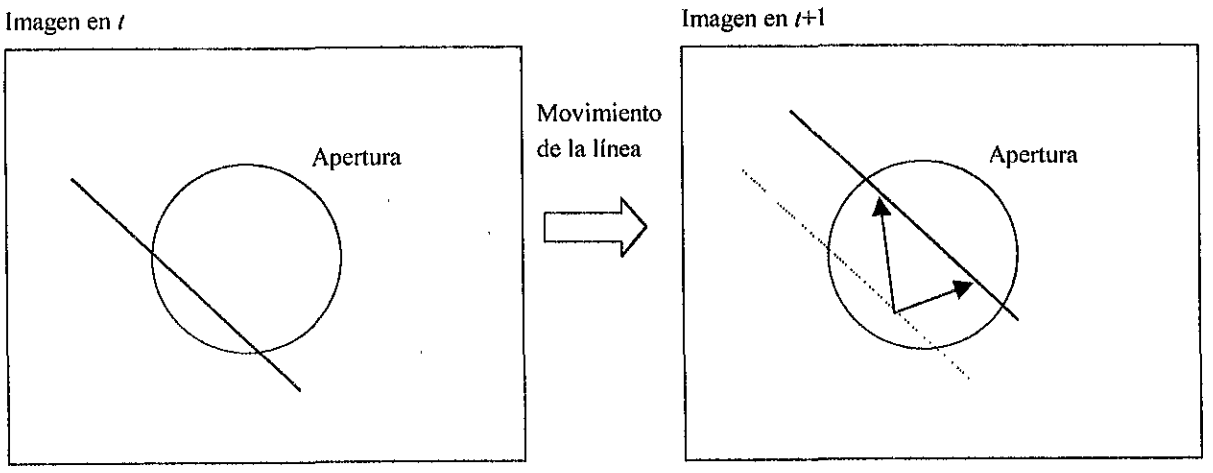


Figura 4.7. El problema de la apertura.

A la situación anterior se le conoce comúnmente como el problema de apertura. Segundo, y el más importante, los estimadores de movimiento basados en la restricción de conservación de los datos son muy sensibles a imágenes ruidosas, especialmente en regiones donde existe una ligera variación espacial, o de textura.

Para solucionar estos problemas, un gran número de esquemas han explotado la restricción de coherencia espacial:

Los puntos vecinos en la escena típicamente pertenecen a la misma superficie y por lo tanto tienen velocidades similares. Ya que los puntos vecinos en la escena proyectan puntos vecinos en el plano imagen, se espera que el flujo óptico varíe suavemente.

Esto se ilustra en la figura 4.8, en donde se ha asumido que los puntos vecinos en la imagen son los mismos que pertenecen a la superficie.

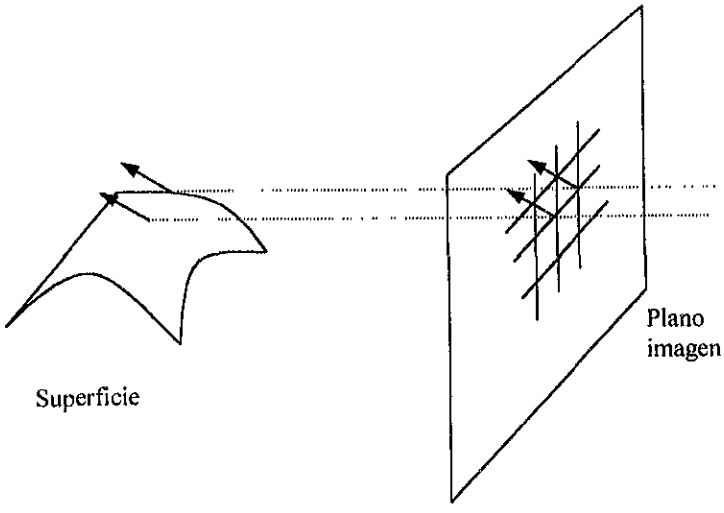


Figura 4.8. Suposición de coherencia espacial.

4.3.3. Persistencia temporal

Las dos restricciones anteriores son comúnmente empleadas en la recuperación del flujo óptico entre dos cuadros en una secuencia de imágenes. Una suposición menos explotada es la continuidad temporal:

El movimiento sobre la imagen de un parche superficial cambia gradualmente con el tiempo

Esta restricción, ilustrada en la figura 4.9 puede ser formulada para tomar en cuenta diversos tipos de movimiento en las imágenes; por ejemplo velocidad o aceleración constantes en el plano imagen.

La continuidad temporal es una restricción poderosa en el sentido que refleja la estabilidad y persistencia de la escena. Muchos intentos para explotar la restricción se han centrado en el procesamiento por lotes de un bloque de imágenes espacio-temporal; por ejemplo, filtrado espacio-temporal [21] y análisis del plano epipolar [25]. Sólo recientemente el poder de la restricción se ha explotado para integrar la información del movimiento sobre el tiempo. La integración temporal mejora los estimadores de movimiento al reducir el ruido sobre el tiempo y reduce los cálculos necesarios para cada nuevo cuadro.

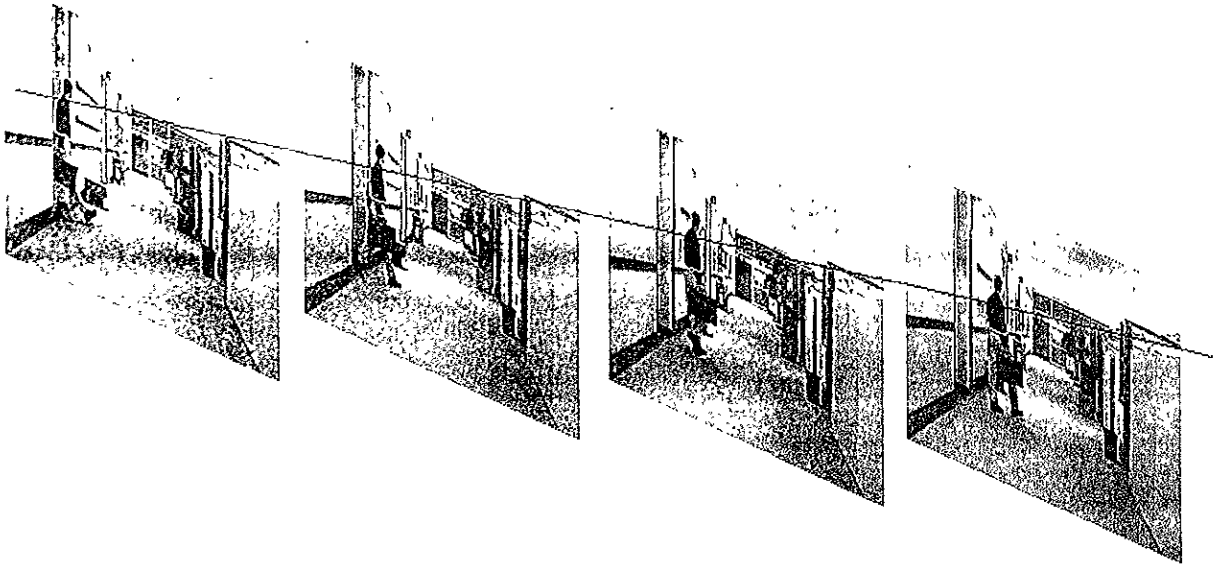


Figura 4.9. Suposición de continuidad temporal.

4.4. Técnicas para la estimación del flujo óptico

Un gran número de técnicas para la recuperación del flujo óptico explotan dos restricciones del movimiento en imágenes: *conservación de los datos* y

coherencia espacial. La restricción en la conservación de los datos se deriva a partir de la observación de que las superficies generalmente persisten en el tiempo, y por lo tanto, la estructura en la intensidad de una región pequeña sobre la imagen permanece constante en el tiempo, aunque su posición puede variar. Esta suposición es formulada frecuentemente como restricciones de primer orden o segundo orden sobre los gradientes en la imagen. Alternativamente, el esquema de correlación intenta encontrar el desplazamiento que minimiza la disparidad entre una región en la imagen y la región desplazada en una imagen posterior, bajo cierto criterio de emparejamiento; por ejemplo, la minimización de la suma de las diferencias al cuadrado entre los pixeles de la región. En muchas situaciones comunes, tal suposición es violada para algún subconjunto de pixeles en la región; por ejemplo, ésta es violada para las fronteras del movimiento y cuando se presentan reflexiones especulares. En tales casos, la restricción en la conservación de los datos continua proporcionando información útil si los puntos violados pueden ser detectados y removidos por consideración. Cuando se presentan cambios globales en el contraste, la simple formulación de la restricción puede proporcionar información poco útil y las mediciones de conservación en los datos pueden considerarse sospechosas.

La restricción de coherencia espacial contempla la suposición que las superficies tienen extensión espacial y por lo tanto los pixeles vecinos en una imagen probablemente pertenecen a la misma superficie. Ya que el movimiento de los puntos vecinos en una superficie rígida y suave cambia gradualmente, se puede aplicar una *restricción de lisado* en el movimiento de pixeles vecinos en el plano imagen.

Esta sección revisa la formulación de la restricción en la conservación de los datos y los tres enfoques principales para explotar la restricción de coherencia espacial: técnicas de regresión, correlación y lisado explícito. También hacemos mención a las suposiciones subyacentes que adoptan los enfoques, indicando cuando son violadas, describiendo los problemas que resultan, y una reseña de los esquemas.

4.4.1. Restricción en la conservación de los datos

Esta sección revisa las suposiciones subyacentes a la mayoría de los algoritmos para la recuperación del flujo óptico. Sea $I_t(\mathbf{p})$ o $I_t(x, y)$ la intensidad de la imagen en el punto $\mathbf{p}=(x, y)$ en el tiempo t. La restricción en la conservación de los datos puede expresarse en términos de la *suposición de intensidad constante* estándar como sigue:

$$\begin{aligned} I_t(\mathbf{p}) &= I_{t+1}(\mathbf{p}+\mathbf{d}) \\ I_t(x, y) &= I_{t+1}(x+dx, y+dy) \\ I_t(x, y) &= I_{t+\delta t}(x+u\delta t, y+v\delta t) \end{aligned} \tag{4.14}$$

en donde $\mathbf{u}=(u, v)$ es la velocidad de imagen vertical y horizontal en un punto y δt es un incremento pequeño en el tiempo. Esto simplemente muestra que el

valor de la imagen en el tiempo t, en el punto **p**, es el mismo que en la siguiente imagen en una localidad desplazada por el flujo óptico.

Los enfoques basados en el cálculo de gradientes, proceden tomando la expansión en serie de Taylor del lado derecho de la ecuación 4.14, resultando:

$$I_t(x, y)=I_t(x, y)+I_x u\delta t+ I_y v\delta t+ I_t\delta t+\in \tag{4.15}$$

en donde I_x, I_y, e I_t, son las primeras derivadas parciales de la intensidad I con respecto a x, y y t respectivamente, y $\in$ contiene los términos de orden superior. Simplificando y dividiendo por δt obtenemos la ecuación de restricción en el flujo óptico estándar.

$$I_x u+ I_y v+ I_t= \nabla I^T \mathbf{u}+ I_t =0 \tag{4.16}$$

Para recuperar un estimado del flujo óptico en un punto, simplemente uno puede minimizar el término de conservación de los datos:

$$E_D(u, v)=\rho(I_x u+ I_y v+ I_t) \tag{4.17}$$

Cuando $\rho(x)=x^2$ el término corresponde al estimador por mínimos cuadrados descrito en [15]. Como se mencionó anteriormente, la sola restricción en la conservación de los datos no es suficiente para recuperar el flujo óptico debido al problema de apertura y a la sensibilidad al ruido.

Dada la ecuación 4.16, podemos ver claramente el problema. La figura 4.10 muestra que el movimiento que satisface la ecuación 4.16 está restringido sólo a una línea en el espacio (u, v). La ecuación restringe al vector de flujo a la dirección del gradiente en la imagen; esto es *normal* a la orientación de la imagen espacial. Por lo tanto el problema requiere restricciones adicionales para recuperar un flujo óptico único.

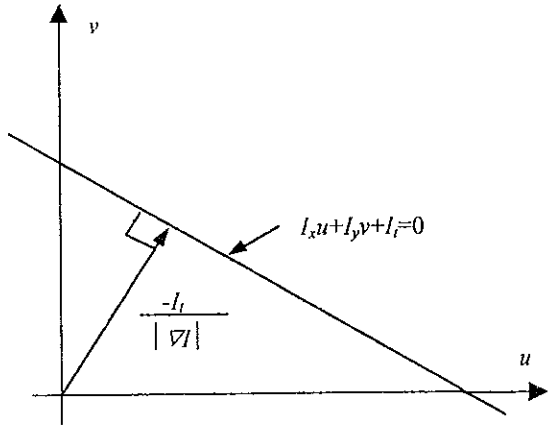


Figura 4.10. Suposición de coherencia espacial.

Suposiciones y violaciones

Existen ciertas suposiciones implícitas subyaciendo la formulación básica que son violadas en la práctica. La simple declaración de constancia en la intensidad y la aproximación de primer orden a la serie de Taylor, asumen

movimiento local de traslación constante y una función de intensidad en la imagen plana. Mientras tales suposiciones son validas en el limite conforme el tamaño de la región se reduce a cero, en la práctica se requiere un tamaño finito para las regiones, y conforme ésta crece la validez de la suposición es más cuestionable. La suposición de constancia en la intensidad también implica que los cambios en intensidad son debidos solamente al movimiento, y por lo tanto, la restricción no toma en cuenta cambios en iluminación, transparencia, o reflexiones especulares.

En la práctica, la restricción impone una suposición de flujo constante sobre un vecindario, Esto resulta del hecho que para estimar derivadas espaciales a partir de imágenes discretas, uno necesita examinar una región en la imagen. Las derivadas espaciales y temporales pueden estimarse utilizando cualquier esquema; por ejemplo diferencias de imágenes, filtrado espacio-temporal.

Sin importar el esquema, los estimados involucran el agrupamiento de información espacial. Para vecindarios pequeños (por ejemplo, diferencias locales), las derivadas estimadas de la imagen son altamente sensibles al ruido, particularmente en áreas con textura. Un enfoque común entonces consiste en usar filtros derivativos que expandan vecindarios grandes. La suposición de movimiento constante, sin embargo, sólo es una buena aproximación para regiones pequeñas. Conforme la región crece, su movimiento será menos aproximado al movimiento constante, y es más probable que contenga movimientos múltiples. El punto importante a notar es que cuando el vecindario para estimar derivadas en imágenes se esparce en una superficie alrededor, las mediciones resultantes carecen de significado. El mejor flujo **u**, obtenido al minimizar la ecuación de restricción en la intensidad (ecuación 4.17), puede ser incorrecto.

4.4.2. Técnicas de regresión

Asumiendo un modelo de flujo constante dentro de una región podemos combinar información de las ecuaciones de restricción de gradiente vecino para determinar el mejor flujo (u, v) que satisfacen todas las ecuaciones al encontrar el (u, v) que minimiza la suma de las restricciones sobre el vecindario:

$$E_D(u,v) = \sum_{(x,y)\in R} \rho(I_x(x,y)u + I_y(x,y)v + I_t(x,y)) \qquad (4.18)$$

en donde $\rho(x)=x^2$ y R es alguna región en la imagen. En forma general, se puede asumir una forma más compleja del modelo de flujo:

$$\mathbf{u}(x, y)=\mathbf{u}(x, y; \mathbf{a})$$

en donde **a** es un vector de parámetros del modelo. Ejemplos de modelos se pueden consultar en la tabla 4.1. Nuestro objetivo es estimar los parámetros **a** del modelo dentro de la región R al minimizar:

$$E_D(\mathbf{a}) = \sum_R \rho(\nabla I^T \mathbf{u}(\mathbf{a}) I_t) \tag{4.19}$$

En el caso constante el modelo es simplemente el mismo que en la ecuación de arriba:

$$\mathbf{u}(\mathbf{a}) = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a1 \\ a2 \end{bmatrix} \tag{4.20}$$

Para un modelo afín tenemos:

$$\mathbf{u}(x, y; \mathbf{a}) = \begin{bmatrix} u(x, y) \\ v(x, y) \end{bmatrix} = \begin{bmatrix} a_1 + a_2 x + a_3 y \\ a_4 + a_5 x + a_6 y \end{bmatrix} \tag{4.21}$$

Note que cuando $\rho(x)=x^2$ es la regresión por mínimos cuadrados estándar. Esta regresión es comúnmente aplicada al apareamiento estéreo [22].

Suposiciones, violaciones, y enfoques previos

El enfoque asume que el movimiento dentro de una región puede ser descrito por un modelo sencillo paramétrico. Cuando una superficie sencilla está presente, el modelo de flujo afín se ha mostrado como una aproximación razonable en muchos casos. Pero conforme crece la complejidad del modelo (esto es, más parámetros deben estimarse), se requieren regiones más grandes para estimaciones exactas. Mientras más grande sea la región bajo consideración, más probable es que contenga movimientos múltiples, y por lo tanto, no sea bien aproximada por el modelo.

Algunas veces es posible detectar movimientos múltiples al examinar el residuo de la solución obtenida por mínimos cuadrados. Esta idea ha sido explotada para producir técnicas iterativas para detectar movimientos múltiples dentro de una región. El esquema es calcular primero el mejor estimador por mínimos cuadrados, y entonces re-calcular el estimador cuadrático y repetir. Tal enfoque es una forma de estimación robusta y ha demostrado trabajar bien cuando el movimiento distractor ocupa una parte pequeña de la región.

El caso de movimientos transparentes múltiples es más complejo. Un enfoque usa un algoritmo iterativo para estimar el movimiento, desarrolla una operación de nulificación para remover los patrones de intensidad que originan el movimiento, y entonces resuelve para el segundo movimiento. El proceso se repite y el estimador de movimiento se refina.

Para utilizar el enfoque por regresión en la estimación de movimiento local, el tamaño de las regiones se debe mantener pequeño para conservar la eficiencia y para reducir la probabilidad de movimientos múltiples. Intuitivamente, esto hace al esquema más sensible al ruido. Un enfoque a considerar es representar explícitamente la incertidumbre en la estimación del flujo. El estimador de incertidumbre puede ser explotado por otros algoritmos más exactos en cuanto a mediciones de movimiento en las imágenes.

A diferencia de las técnicas de regresión anteriores que intentan encontrar el mejor flujo dada la ecuación de restricción en iluminación, Schunk [25] propone un método de *agrupamiento por restricción de línea* para el cálculo de estimadores de flujo cerca de las discontinuidades de movimiento. El enfoque, que desempeña un análisis de agrupamiento en la intersección de líneas de restricción dentro de una vecindad, puede ser visto como una técnica estadística robusta. Mientras el esquema no formula el problema de flujo óptico en términos de una estimación robusta, Schunk sugiere que "futuros experimentos deben ser conducidos a comparar estimaciones robustas contra el agrupamiento por restricción de línea". Schunck también ha usado la técnica de regresión por mínimos cuadrados para determinar robustamente la mejor intersección por restricción de línea dentro de un vecindario.

4.4.3. Técnicas de correlación

Las técnicas de correlación son similares a los esquemas de regresión, en el sentido que empiezan con la consideración de intensidad constante, pero a diferencia de la formulación del gradiente, adoptan una estrategia de emparejamiento. Dada una región en una imagen el objetivo es encontrar el desplazamiento (u, v), de aquella región en la siguiente imagen que minimiza el siguiente error:

$$E_D(u,v) = \sum_{(x,y)\in R} [I_t(x,y) - I_{t+\delta t}(x+u\delta t, y+v\delta t)]^2 \qquad (4.22)$$

Esta es la medida a partir de la suma de diferencias cuadradas estándar (SSD Sum of Squared Differences). Tomando la expansión en serie de Taylor de la formulación de correlación, resulta exactamente la misma formulación que el esquema de regresión. Adicionalmente, se ha demostrado que, conforme δt tiende a cero, el resultado de minimizar la formulación SSD converge con aquel obtenido por el esquema basado en el gradiente de segundo orden. También se puede demostrar que conforme el tamaño de R tiende a cero, su formulación SSD converge a la formulación basada en el gradiente de primer orden.

Note que, sobre un rango de desplazamiento, (u, v), la ecuación da origen a una superficie de correlación en donde el mínimo corresponde al mejor desplazamiento dado el criterio de emparejamiento. Esta superficie de correlación, E_D, usualmente se calcula sobre desplazamientos discretos. Determinar el mejor emparejamiento para éste caso es sencillo, pero no proporciona exactitud subpixel. Uno puede interpolar la imagen y calcular E_D en desplazamientos subpixel, pero implica un problema de búsqueda extensivo y la interpolación introduce suposiciones acerca de la estructura subyacente. En forma más común, se calcula la superficie SSD discreta, se encuentra el mejor desplazamiento, y se calcula la estimación subpixel al ajustar un cuadrático a un mínimo.

La correlación es popular en visión por computadora y ha formado la estrategia básica de emparejamiento en muchos algoritmos de movimiento y visión estéreo. También se ha utilizado para seguimiento, y el hardware de correlación en tiempo real la hace atractiva para aplicaciones de robótica.

Suposiciones, violaciones, y enfoques previos

El esquema de correlación asume que el campo de flujo puede ser aproximado por un movimiento traslacional uniforme dentro de la región de interés. En la práctica, pequeños aumentos de rotación, divergencia y ocultamientos pueden tolerarse. Mientras el esquema asume iluminación constante, algunos cambios de iluminación pueden ser acomodados al utilizar una forma normalizada de correlación o al pre-procesar mediante filtrado laplaciano.

Existe un acuerdo que debe ser abordado al usar esquemas de correlación: conforme el tamaño de la ventana de correlación aumenta para mejorar la confiabilidad del estimador de movimiento, crece la probabilidad de que movimientos múltiples contaminen la solución. Para lidiar con movimientos múltiples dentro de una ventana, Okutomi y Kanade desarrollaron una técnica de "ventana adaptable" que ajusta el tamaño de la ventana de correlación para minimizar la incertidumbre de la estimación. La implementación del esquema está limitado por el uso de una ventana de forma fija (rectangular) que no se puede adaptar a fronteras de forma arbitraria. El esquema tampoco puede lidiar con los problemas de fragmentos ocluidos, en donde, sin importar el tamaño o forma de la ventana, se presentan movimientos múltiples. Sería interesante extender el enfoque de ventana adaptable para permitir que un subcojunto arbitrario de pixeles dentro de una región sea no considerado.

Cuando existen movimientos múltiples dentro de una región de correlación, la superficie de correlación puede contener múltiples mínimos correspondientes a diferentes movimientos, Adicionalmente, los movimientos pueden interferir entre ellos, haciendo que los mínimos estén menos definidos, y por lo tanto, más sensibles al ruido y menos detectables realistamente. En áreas de baja textura, el ruido puede producir mínimos múltiples en la superficie de correlación. Los esquemas basados en confidencia intentan tratar con violaciones del término de datos al asignar bajas confidencias a éstas mediciones. Por ejemplo, Anandan calcula mediciones de confidencia selectivas y direccionales basadas en la SSD. Las áreas de baja textura no dan origen al crecimiento de picos agudos en la superficie SSD y por lo tanto son asignadas con bajas confidencias. Por lo tanto, las áreas más sensibles al ruido reciben bajas confidencias.

4.4.4. Técnicas de lisado explícito

Las técnicas anteriores basadas en área emplean una restricción de coherencia espacial implícita en el sentido que el flujo dentro de una región se asume que conforma un modelo de movimiento sencillo. Esta sección lidia con

esquemas de regularización que explícitamente implementan la restricción de coherencia espacial. La noción de restricción de lisado es motivada por el hecho de que la información local de gradiente sólo puede restringir parcialmente la solución. Adicionalmente, las mediciones de gradiente local son sensibles al ruido, particularmente en áreas que contienen ligeras variaciones en contraste. La introducción a la restricción de coherencia espacial restringe la clase de soluciones admisibles, resultando en un problema bien planteado.

El término de conservación de los datos, E_D, se combina con los términos de lisado explícito, E_S, para formar una función objetivo, E, la cual debe ser minimizada:

$$E(\mathbf{u})=\lambda E_D(\mathbf{u})+E_S(\mathbf{u}) \tag{4.23}$$

en donde λ controla la importancia relativa de los dos términos. El término de lisado se define como una restricción local sobre un vecindario espacial pequeño. En éste sentido es conveniente adoptar la formulación basada en Campos Aleatorios de Markov (Markov Random Fields MRF). En esta sección, la formulación introducida es puramente por conveniencia notacional y los MRF no son abordados en detalle.

Para una imagen de tamaño nXn pixeles definimos un enrejado de *posiciones* (figura 4.11):

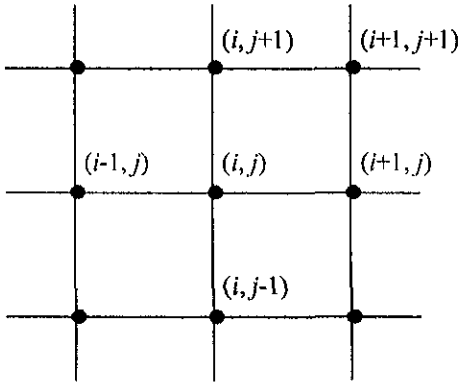


Figura 4.11. Una imagen es tratada como un enrejado de *posiciones*.

$$S=\{s_1, s_1, \ldots s_{n2} \mid \forall w\ 0\le i(s_w), j(s_w)\le n-1\}$$

en donde $(i(s_w), j(s_w))$ denota las coordenadas del pixel con posición s. Para formulaciones diferentes de la restricción, se pueden definir diferentes *sistemas de vecindad*, $\mathcal{G}$, que determinan la interacción local de las posiciones. Un sistema de vecindad, $\mathcal{G}=\{\mathcal{G}_s, s\in S\}$, satisface las siguientes condiciones:

1. $\mathcal{G}_s \subseteq S$
2. $s \notin \mathcal{G}_s$
3. $s\in\mathcal{G}_t \Leftrightarrow t\in \mathcal{G}_s$

El par $\{S, \mathcal{G}\}$ define una gráfica con $s \in S$ representando los vértices y pares, $\{(s,t) \mid s \in \mathcal{G}_t\}$, siendo los bordes. Definimos como una camarilla al conjunto de posiciones , $C \subseteq S$, tal que si s, $t \in C$ y $s \neq t$, entonces $t \in \mathcal{G}_s$. Sea $\mathcal{C}$ un conjunto de camarillas.

En el caso de imágenes, estamos interesados en tipos particulares de gráficos, llamados enrejados. Por lo tanto consideramos sistemas locales de vecindario de la forma:

$$\mathcal{G}_s = \{t \mid 0 < (i(s)-i(t))^2 + (j(s)-j(t))^2 \leq c\}$$

La figura 4.12 muestra los sistemas de vecindario para varios valores de c. Para las restricciones de primer orden $(c=1)^4$, sólo buscamos por las relaciones de vecindad más cercanas (Norte, Sur, Este, Oeste) en el enrejado:

$$\mathcal{G}_s = \{t \mid (i,j)=(i(s),j(s)),(i(t),j(t)) \in \{(i+1,j),(i,j+1),(i-1,j),(i,j-1)\}\}$$

Aún éste sistema de vecindario simple es útil y tiene el beneficio añadido de ser realizado fácilmente en arquitecturas paralelas.

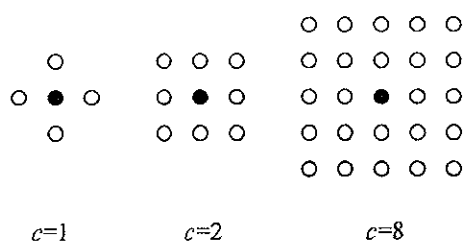


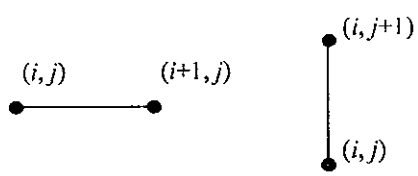
Figura 4.12. Varios tamaños de vecindades en un enrejado.

Modelo de flujo constante

La más común de las formulaciones de E_S es el modelo de *primer orden*. Se toma como medida de lisado el cuadrado del gradiente del campo de velocidad:

$$E_S(u,v) = u_x^2 + u_y^2 + v_x^2 + v_y^2 \tag{4.24}$$

en donde los subíndices indican las derivadas parciales en la dirección x o y. Se puede aproximar ésta ecuación utilizando el sistema discreto de vecindad de primer orden en donde se consideran los siguientes vecinos:



entonces resulta la siguiente aproximación:

$$E_S(\mathbf{u}_{i,j}) = (u_{i,j}-u_{i+1,j})^2 + (u_{i,j}-u_{i,j+1})^2 + (v_{i,j}-v_{i+1,j})^2 + (v_{i,j}-v_{i,j+1})^2 \tag{4.25}$$

El mínimo de esto es simplemente el flujo medio $\bar{\mathbf{u}}$ de los puntos vecinos al Norte y Este:

$$\frac{\partial Es}{\partial u} = u - \frac{1}{2}(u_{i+1,j} + u_{i,j+1}) = u - \bar{u} \tag{4.26}$$

$$\frac{\partial Es}{\partial v} = v - \frac{1}{2}(v_{i+1,j} + v_{i,j+1}) = v - \bar{v} \tag{4.27}$$

Note que el flujo medio es el mejor estimador por mínimos cuadrados del flujo para un modelo de flujo constante. Entonces, el modelo simple de primer orden implica un campo de flujo óptico local constante. Un estimador más realizable del flujo medio puede obtenerse al considerar una región más grande. Por ejemplo, Horn sugiere calcular la media al convolucionar los componentes del flujo con la máscara:

$$\frac{1}{20}\begin{bmatrix} 1 & 4 & 1 \\ 4 & 0 & 4 \\ 1 & 4 & 1 \end{bmatrix}$$

Siguiendo el esquema de Horn y Schunck [25], se puede tomar a E_D como la ecuación de restricción de gradiente y E_S como el modelo simple de flujo constante. Esto da origen a la siguiente formulación por mínimos cuadrados del flujo óptico:

$$E(\mathbf{u}) = \lambda(I_x u_{i,j} + I_y v_{i,j} + I_t)^2 + \frac{1}{2}\left[(u_{i,j} - u_{i,j})^2 + (v_{i,j} - v_{i,j})^2\right] \tag{4.28}$$

Esta formulación admite un esquema simple de relajación iterativa para determinar el flujo óptico:

$$u_{i,j}^{(n+1)} = u_{i,j}^{n} - \frac{I_x(I_x \bar{u}_{i,j}^{n} + I_y \bar{v}_{i,j}^{n} + I_t)}{1 + \lambda(I_x^2 + I_y^2)}$$

$$v_{i,j}^{(n+1)} = \bar{v}_{i,j}^{n} - \frac{I_y(I_x \bar{u}_{i,j}^{n} + I_y \bar{v}_{i,j}^{n} + I_t)}{1 + \lambda(I_x^2 + I_y^2)} \tag{4.29}$$

Intuitivamente, el problema con los esquemas simples de relajación como el utilizado es que para reducir los efectos del ruido uno debe lisar el campo de flujo. Permaneciendo fiel a las mediciones de la imagen resulta en un campo de flujo ruidoso. Lo que se necesita es una forma de ignorar mediciones ruidosas y al mismo tiempo evitar lisado a través de las discontinuidades.

Modelo de flujo afín

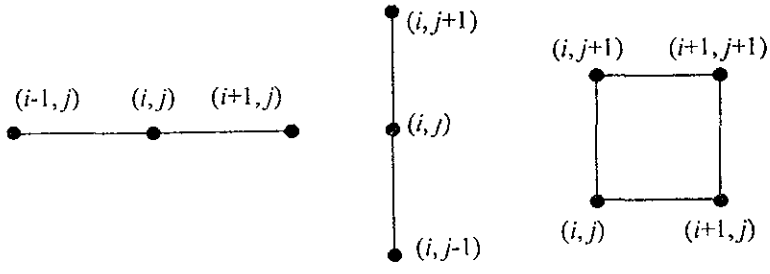
Ahora consideremos la restricción de lisado de segundo orden. Por simplicidad de notación se considera sólo la componente horizontal del flujo,

u; el tratamiento es idéntico para la componente vertical. La restricción de segundo orden es:

$$E_S(u)=u_{xx}^2+2u_{xy}^2+u_{yy}^2 \tag{4.30}$$

en donde los subíndices indican las segundas derivadas parciales.

Utilizando el siguiente sistema de vecindad:



a *b* *c*

el problema sé discretiza de la siguiente forma:

$$u_{xx}=u_{i,j-1}+u_{i,j+1}-2u_{i,j} \tag{4.31}$$

$$u_{yy}=u_{i-1,j}+u_{i+1,j}-2u_{i,j} \tag{4.32}$$

$$u_{xy}=-u_{i,j}-u_{i+1,j+1}+u_{i,j+1}+u_{i+1,j} \tag{4.33}$$

La restricción de lisado es minimizada cuando las segundas derivadas parciales u_{xx}, u_{yy}, y u_{xy}. Este es el caso cuando el campo de flujo es localmente *afín*: esto es, lineal en x y y. Tales modelos afines del flujo óptico se han vuelto populares en formulaciones basadas en regresión como una alternativa al modelo de flujo constante.

Suposiciones, violaciones, y enfoques previos

Los dos modelos de lisado descritos anteriormente ambos asumen un modelo sencillo que describe localmente el flujo óptico. Considere que pasa si el campo de flujo es discontinuo; es decir, existen movimientos múltiples presentes en el vecindario. La figura 4.13 ilustra la situación. La aproximación a flujo constante forza a $u_{i,j}$ a promediarse con sus vecinos $u_{i-1,j}$, $u_{i,j+1}$, $u_{i-1,j}$, y $u_{i,j-1}$. El promediado de la restricción en el lisado resultará en un *manchado* a través de la frontera de movimiento. No sólo esto reduce la exactitud del campo de flujo, pero esto oscurece la información estructural importante acerca de la presencia de una frontera objeto. En lugar de sobre-lisar, lo que a uno le gustaría hacer es que el flujo en $u_{i+1,j}$ sea diferente del resto e ignorarlo.

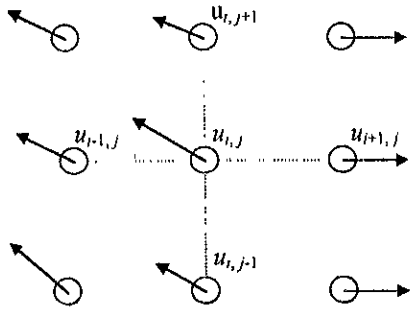
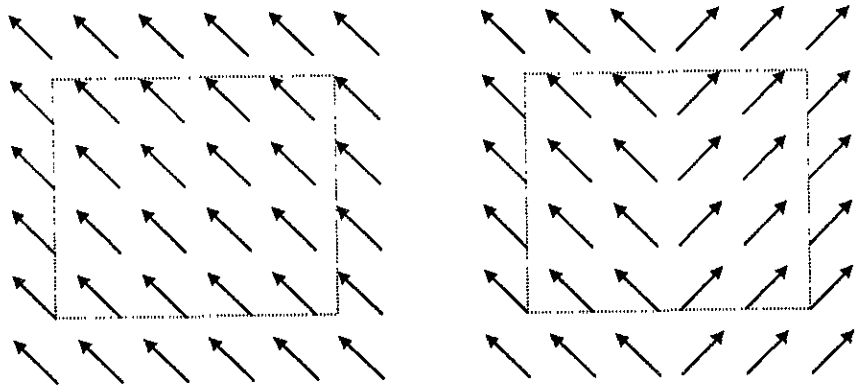


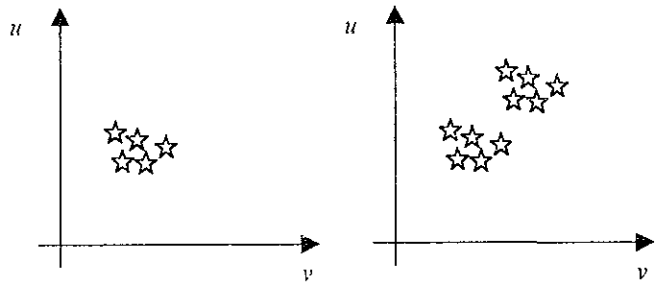
Figura 4.13. Lisado a través de las discontinuidades de flujo.

Otra forma de ver el problema es al considerar la distribución de los vectores de flujo sobre un vecindario grande. Por ejemplo, el vecindario en la figura 4.14.a) contiene un conjunto de vectores de flujo que son considerados bajo una suposición de flujo constante. En contraste, en el vecindario de la figura 4.14.b) se extiende una frontera de movimiento; los vectores de flujo en la región yacen sobre dos grupos distintos. Esto se puede observar en la figura 4.15 en donde los vectores de flujo dentro del vecindario están graficados en un sistema coordenado *u-v*. La figura 4.15.a) corresponde al flujo constante dentro de la región; en esta situación, los vectores son agrupados en el espacio *u-v*, y el flujo promedio proporciona un estimado aproximadamente razonable del movimiento. La figura 4.15.b) corresponde al caso de movimiento múltiple en donde los vectores de flujo forman grupos múltiples y distintos en el espacio *u-v*.



a) b)

Figura 4.14. Vecinos locales de los vectores de flujo; **a).** Movimiento sencillo dentro de un vecindario, **b)** movimiento múltiple dentro de un vecindario



a) b)

Figura 4.15. *Distribuciones locales de los vectores de flujo;* **a)** movimiento sencillo. **b)** movimiento múltiple.

En el caso de grupos múltiples, el flujo medio no desempeña un buen trabajo al caracterizar el flujo de cada grupo. Al contrario, en casos como estos, el objetivo debe ser encontrar el flujo que mejor describa a la mayoría de los datos. Existen numerosas técnicas que la emplean; la más importante es el esquema de *aprovechamiento en línea* que proporciona técnicas generales para la regularización de discontinuidades. Existen otras técnicas que incluyen reglas heurísticas en donde se explota la información acerca de la intensidad en la imagen, como también algoritmos que intentan detectar discontinuidades en el campo de flujo.

4.5. Estimación de parámetros

El problema de encontrar la correspondencia entre dos proyecciones planas diferentes (imágenes en t y en t+1, Figura 4.16) del mismo objeto en el espacio 3D, se puede simplificar al asumir parches planos. Entonces el problema se puede plantear como la identificación del vector de desplazamiento $\mathbf{d}(\Theta)$, bajo la restricción de flujo óptico, es decir, esencialmente un problema de minimización.

Si bien en situaciones reales el objeto 3D permanece estático, el fenómeno se puede plantear como un objeto en movimiento 3D proyectando su situación en el espacio para dos instantes de tiempo.

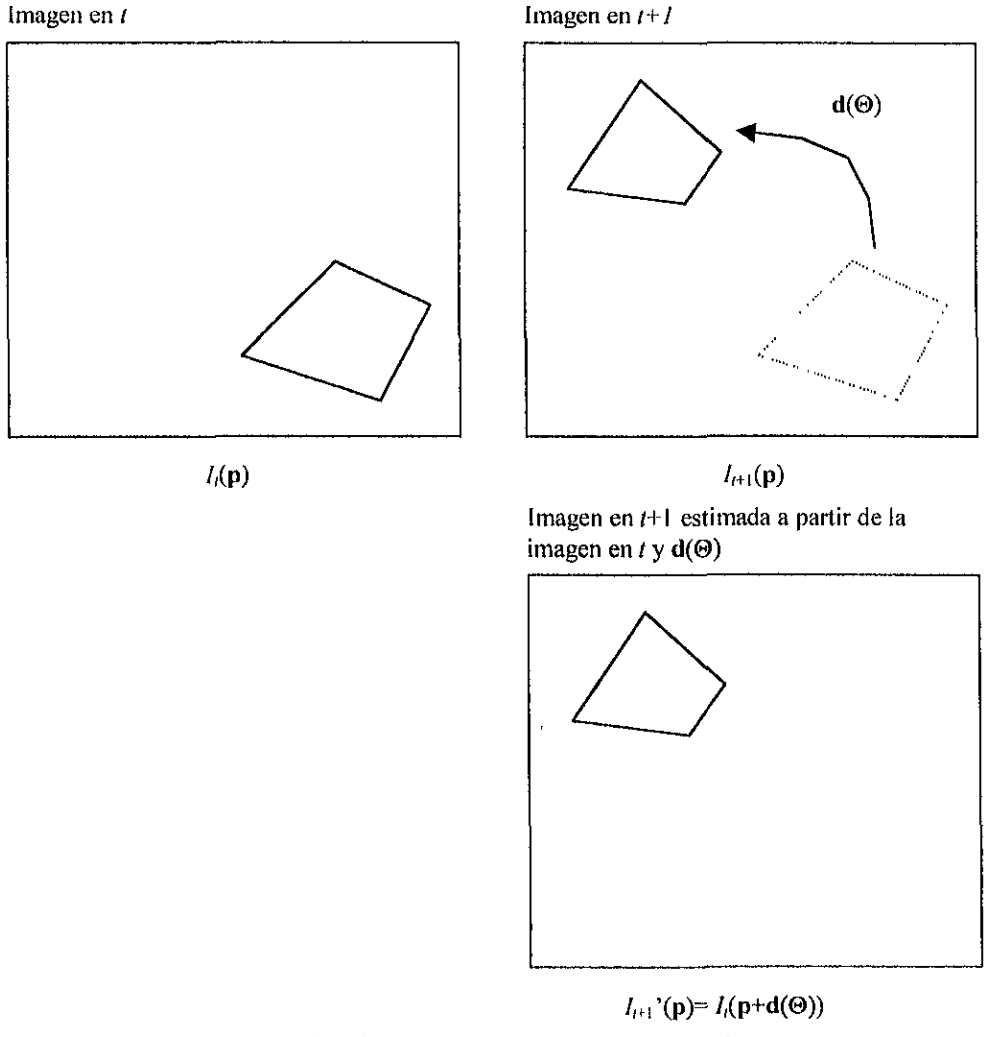


Figura 4.16. Estimación de movimiento.

4.5.1. Modelo de movimiento

En éste enfoque, se proyecta el movimiento físico 3D en un plano de imagen. Asumiendo que los objetos en el espacio sufren translación, rotación y deformación lineal, el movimiento puede ser descrito por (ver figura 4.17):

$$\mathbf{q}' = \mathbf{R}\,\mathbf{q} + \mathbf{T} \tag{4.34}$$

en donde

$$\mathbf{q}' = \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} \quad \mathbf{q} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} r_{11} r_{12} r_{13} \\ r_{21} r_{22} r_{23} \\ r_{31} r_{32} r_{33} \end{bmatrix} \quad \mathbf{T} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$$

en donde $\mathbf{q}'$ es el vector de coordenadas desplazadas, $\mathbf{q}$ es el vector de coordenadas original, $\mathbf{R}$ es la matriz de rotación y deformación lineal, y $\mathbf{T}$ es el vector de traslación.

La proyección plana bajo el modelo de cámara pin hole con longitud focal F es del tipo (ver figura 4.17)

$$x = F\frac{X}{Z} \qquad y = F\frac{Y}{Z} \qquad x' = F\frac{X'}{Z'} \qquad y = F\frac{Y'}{Z'} \tag{4.35}$$

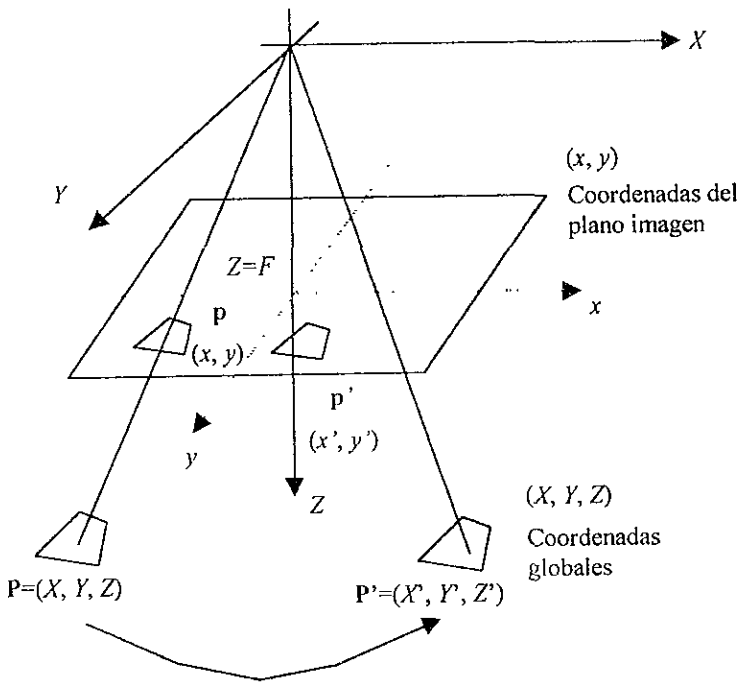


Figura 4.17. Proyección plana de un objeto 3D.

Sustituyendo la ecuación 4.35 en 4.34:

$$x' = F\frac{r_{11}x + r_{12}y + Fr_{13} + FT_x / Z}{r_{31}x + r_{32}y + Fr_{33} + FT_z / Z}$$

$$y' = F\frac{r_{21}x + r_{22}y + Fr_{23} + FT_y / Z}{r_{31}x + r_{32}y + Fr_{33} + FT_z / Z} \tag{4.36}$$

Para una descripción simple y económica del movimiento 2D proyectado, consideremos que el movimiento del objeto está restringido a traslación y rotación 2D y cambio de escala en el plano imagen. Entonces la versión simplificada de (4.36) es

$$x' = \lambda x - \alpha y + T_x$$

$$y' = \alpha x + \lambda y + T_y \tag{4.37}$$

en donde se define el vector de parámetros $\Theta^T = [Tx\ Ty\ \alpha\ \lambda]$ ó

$$d(\Theta) = [\lambda x - \alpha y + T_x\ ,\ \alpha x + \lambda y + T_y]$$

4.5.2. Estimación cuadrática y robusta

Consiste en estimar los parámetros Θ que proporcionan la correspondencia en un par de imágenes. El algoritmo utilizado es conocido como "pel recursivo", en donde se minimiza la Diferencia del Conjunto Desplazado (DFD, Displaced Frame Diference):

La restricción de conservación de intensidad constante puede plantearse como:

$$I_t(\mathbf{p})=I_{t+1}(\mathbf{p}+\mathbf{d}) \tag{4.38}$$

en donde

$\mathbf{p}=(x, y)$	vector de coordenadas
$\mathbf{d}=(dx, dy)$	vector de desplazamiento
$dx=tx+k(x-xg)-\theta(y-yg)$	desplazamiento en la dirección x
$dy=ty+k(y-yg)+\theta(x-xg)$	desplazamiento en la dirección y
(xg, yg)	centro de gravedad de la región Ri
$\Theta=(tx, ty, k, \theta)$	vector de parámetros

La ecuación 4.38 se puede resolver para $\mathbf{d}$ mediante la minimización del término de conservación de intensidad constante

$$\begin{aligned} ED(\mathbf{p}, \mathbf{d})&=\rho(I_t(\mathbf{p})-I_{t+1}(\mathbf{p}+\mathbf{d})) \\ ED(\mathbf{p}, \mathbf{d})&=\rho(DFD(\mathbf{p}, \mathbf{d})) \end{aligned} \tag{4.39}$$

en donde:

$DFD(\mathbf{p},\mathbf{d})=I_t(\mathbf{p})-I_{t+1}(\mathbf{p}+\mathbf{d})$	Diferencia de cuadro desplazado
$\rho(z)$	Estimador

Cuando $\rho(z)=z^2$, esto corresponde al estimador estándar por mínimos cuadrados.

Cuando $\rho(z)=\log\left(1+\frac{1}{2}\left(\frac{z}{\sigma}\right)^2\right)$ esto corresponde al estimador lorentziano robusto, en donde σ es la desviación estándar.

Resolviendo la minimización de la ecuación 4.39 por el método del gradiente, nos conduce al cálculo iterativo del vector de parámetros

$$\Theta^{i+1}=\Theta^{i}-\frac{1}{2Nj}\sum_{\mathbf{p}\in Rj}\varepsilon_\beta \mathbf{G}^i \tag{4.40}$$

en donde

Nj Tamaño de la región Rj

ε_β Matriz de ganancia adaptable

$$\varepsilon\beta = \frac{1}{\nabla^2 x I_{t+1}(\mathbf{p}+\mathbf{d}) + \nabla^2 y I_{t+1}(\mathbf{p}+\mathbf{d}) + \alpha} \begin{bmatrix} \varepsilon c 000 \\ 0 \varepsilon c 00 \\ 00 \varepsilon l 0 \\ 000 \varepsilon \end{bmatrix}$$

ε_c=1, ε_l=0.01 Parámetros de ponderación

α=0.001 Valor para evitar la división entre cero

$\mathbf{G}^t$ Matriz de gradientes

$$\mathbf{G}^t = \begin{bmatrix} \frac{\partial}{\partial tx} ED(\mathbf{p},\mathbf{d}) \\ \frac{\partial}{\partial ty} ED(\mathbf{p},\mathbf{d}) \\ \frac{\partial}{\partial k} ED(\mathbf{p},\mathbf{d}) \\ \frac{\partial}{\partial \theta} ED(\mathbf{p},\mathbf{d}) \end{bmatrix}$$

Al resolver las derivadas parciales es fácil demostrar las siguientes soluciones a la ecuación 4.40, para el caso cuadrático y robusto lorenziano, respectivamente

caso cuadrático

$$\begin{bmatrix} tx^{t+1} \\ ty^{t+1} \\ k^{t+1} \\ \theta^{t+1} \end{bmatrix} = \begin{bmatrix} tx^t \\ ty^t \\ k^t \\ \theta^t \end{bmatrix} - \frac{1}{Nj \sum_{\mathbf{p} \in R_j} \nabla^2 x I_{t+1}(\mathbf{p}+\mathbf{d}) + \nabla^2 y I_{t+1}(\mathbf{p}+\mathbf{d}) + \alpha} \bullet \tag{4.41}$$

$$\bullet \begin{bmatrix} \varepsilon c \nabla x I_{t+1}(\mathbf{p}+\mathbf{d}) \\ \varepsilon c \nabla y I_{t+1}(\mathbf{p}+\mathbf{d}) \\ \varepsilon(x-xg)\nabla x I_{t+1}(\mathbf{p}+\mathbf{d}) + \varepsilon(y-yg)\nabla y I_{t+1}(\mathbf{p}+\mathbf{d}) \\ -\varepsilon(y-yg)\nabla x I_{t+1}(\mathbf{p}+\mathbf{d}) + \varepsilon(x-xg)\nabla y I_{t+1}(\mathbf{p}+\mathbf{d}) \end{bmatrix}$$

caso robusto lorenziano

$$\begin{bmatrix} tx^{t+1} \\ ty^{t+1} \\ k^{t+1} \\ \theta^{t+1} \end{bmatrix} = \begin{bmatrix} tx^{t} \\ ty^{t} \\ k^{t} \\ \theta^{t} \end{bmatrix} - \frac{1}{Nj} \sum_{\mathbf{p} \in Rj} [\nabla^2 x I_{t+1}(\mathbf{p}+\mathbf{d}) + \nabla^2 y I_{t+1}(\mathbf{p}+\mathbf{d})] + \alpha \bullet \frac{DFD(\mathbf{p},\mathbf{d})}{\bullet \frac{1}{[2\sigma^2 + DFD^2(\mathbf{p},\mathbf{d})]} \begin{bmatrix} \varepsilon_c \nabla x I_{t+1}(\mathbf{p}+\mathbf{d}) \\ \varepsilon_c \nabla y I_{t+1}(\mathbf{p}+\mathbf{d}) \\ \varepsilon_l (x - xg) \nabla x I_{t+1}(\mathbf{p}+\mathbf{d}) + \varepsilon_l (y - yg) \nabla y I_{t+1}(\mathbf{p}+\mathbf{d}) \\ -\varepsilon_l (y - yg) \nabla x I_{t+1}(\mathbf{p}+\mathbf{d}) + \varepsilon_l (x - xg) \nabla y I_{t+1}(\mathbf{p}+\mathbf{d}) \end{bmatrix}} \tag{4.42}$$

La desviación estándar σ del estimador robusto generalmente se resuelve en paralelo con la estimación de parámetros. Esta cantidad usualmente es

σ=1.48 Med($|$ *Ri*-Med(*Rj*) $|$) en donde *Ri* y *Rj* son regiones adyacentes [16].

4.6. Segmentación espacio temporal

Una segmentación espacial pura es demasiado redundante en el caso de secuencias en movimiento, ya que ciertas regiones espaciales pueden formar una entidad única de movimiento conocida como *macro región espacio-temporal homogénea MR*. De esta manera, regiones espaciales adyacentes pueden fusionarse en regiones espacio-temporales. Por lo tanto, como un criterio de homogeneidad temporal, elegimos el *error cuadrático medio de predicción* (Mean Square Prediction Error MSPE) que se encuentra sobre la base de la diferencia del conjunto desplazado:

$$DFD_i^2 = \frac{1}{N} \sum_{p \in R_i} [I_t(\mathbf{p}) - I_{t+1}(\mathbf{p}+\mathbf{d})]^2$$

Para cada posible fusión de dos regiones adyacentes R_{i1} y R_{j1} en la imagen I_1, se estiman los vectores de parámetros y se elige la mejor fusión (la cual proporciona el mínimo DFD^2) si se satisface el criterio de fusión. El proceso completo de fusión se realiza sobre la base de la fusión de regiones adyacentes bien compensadas [31].

Se dice que una región R_i está bien compensada si su error $DFD^2{}_i$ es menor que un umbral fijo. En nuestro caso, usamos un umbral de comparación μ =20 ya que su correspondiente *PSNR*=35dB caracteriza a las imágenes reconstruidas con defectos no perceptibles. Sea $\{R_j\}$ un conjunto de regiones bien compensadas adyacentes a la región R_i. Para una posible unión $R_i \cup R_j$, el vector óptimo de parámetros de movimiento θ_{ij} es estimado y un conjunto de valores $\{DFD^2{}_{ij}\}$ se obtiene. Una región $R_j \in \{R_j\}$ se fusiona con R_i si

$$DFD^2_{ij}=\min j\ DFD^2_{ij} \text{ y } DFD^2_{ij}<\mu \tag{4.43}$$

En donde μ es el umbral anterior. El proceso es reiterado hasta el agotamiento de las posibilidades. Además:

$$DFD_{ij} = \frac{1}{R_i} \cdot \sum_{\mathbf{p} \in R_i} [I_i(\mathbf{p}) - I(I_{i+1}(\mathbf{p}+\mathbf{d})]^2 \tag{4.44}$$

en donde R_i denota el número de pixeles en la región R_i e I es un interpolador bilineal.

En cuanto al pre-procesamiento de la imagen comprobamos que el filtrado morfológico homogeneiza regiones y preserva contornos al conectar únicamente zonas planas. Por otra parte, la segmentación espacial extrae regiones de interés con menor carga computacional y sobre-segmentación, debido al pre-procesamiento. Para evaluar el desempeño del análisis de movimiento realizamos dos experimentos: primero aislamos una región espacial y estimamos movimiento de forma independiente; posteriormente fusionamos regiones con movimiento similar.

4.6.1. Desempeño del análisis de movimiento

El análisis de movimiento (tabla 4.2) es comparado en desempeño (relación señal a ruido) para una región de la secuencia 38 y 40 de *Miss América* (figura 4.18), utilizando los dos enfoques expuestos: *cuadrático* (ecuación 4.41) y *robusto* (ecuación 4.42), adicionalmente al clásico emparejamiento por bloques (*block matching*). Las secuencias originales de *Miss América* tienen un psnr inherente al movimiento de 12.04.

No obstante que la calidad de la imagen reconstruida es aparentemente superior para el caso del estimador cuadrático, la coherencia de los parámetros de movimiento es otro punto que debe tomarse en cuenta. En éste sentido, existen dos argumentos que justifican el uso del enfoque robusto. El primero es que un número reducido de parámetros es suficiente para describir completamente los vectores de parámetros, que pueden ser grandes, y tales vectores constituir una aproximación más coherente al flujo óptico real. El segundo argumento es el costo en tiempo de cálculo.

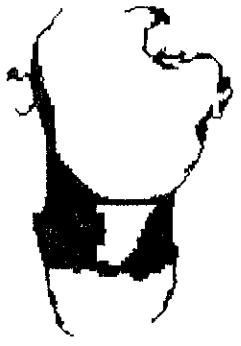


Figura 4.18. Región de la secuencia *Miss América*.

Parámetro	Block matching	Estimador cuadrático	Estimador robusto
tx	-6.3476	-6.3871	-6.3371
ty	-1.2057	-1.2129	-1.2164
k	-	-0.0019	-0.0010
θ	-	0.0013	-0.0034
psnr	27.6	29.25	27.96

Tabla 4.2. Comparación para el análisis de movimiento.

4.6.2. Fusión de regiones por movimiento

La etapa anterior de segmentación espacial aún produce una imagen sobre-segmentada. Si bien sigue el patrón básico de intensidades, ésta tiende a crear particiones adyacentes con propiedades de movimiento similares. En éste sentido y bajo el esquema de compresión de video, es necesario fusionar regiones con movimiento similar, con el objetivo de lograr bajas tasas de transmisión.

Una manera eficiente de fusionar se asemeja bastante a la utilizada para la segmentación espacial. Nuestro algoritmo trabaja al seleccionar regiones pivote aleatoriamente y examinar si sus vecinos son similares en movimiento (ecuación 2.44) para fusionarlos. El proceso trabajando aleatoriamente, puede fusionar eventualmente regiones que en un primer intento no lo fueron. El desempeño anterior es posible ya que los parámetros de movimiento que describen a regiones aisladas tiende a adaptarse a una parámetro regional, conforme otras regiones adyacentes van fusionándose.

En ésta sección adoptamos el criterio de similitud para la fusión de regiones, es decir, la segmentación espacio temporal, al considerar parámetros descriptores de movimiento para cada región dividida, y fusionar sobre la base de la similitud de descriptores. Posteriormente planteamos el algoritmo para la fusión. A continuación presentamos un guía como ejemplo de implementación orientado a la programación en lenguaje C y finalmente un ejemplo práctico sobre la secuencia estándar *Miss América*.

Descripción del algoritmo

El algoritmo trabaja con base a la selección de una región pivote R_i en forma aleatoria. Entonces se examinan los vecinos R_{ij} mediante el enunciado lógico para la fusión:

$$\text{Si } \text{abs}_{\min}(DFD^2_{ij} - DFD^2_{ii}) < \mu \text{ entonces } R_i,\ R_{ij} \text{ se fusionan.}$$

$$\text{si } \text{abs}_{\min}(DFD^2_{ij} - DFD^2_{ii}) \geq \mu \text{ entonces } R_i,\ R_{ij} \text{ no se fusionan.} \quad (4.45)$$

Para la siguiente discusión refiérase a la figura 4.19.

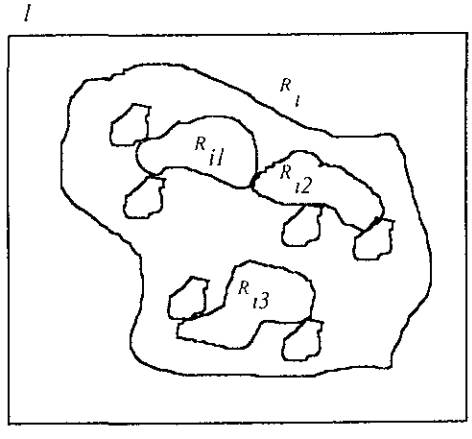


Figura 4.19. Descripción del algoritmo para la fusión por movimiento.

Asumamos que R_i es un conjunto de regiones con parámetro de movimiento global Θ_{Ri} de la imagen I_t, la cual es planteada como un posible resultado de la segmentación espacio-temporal. Las regiones R_{i1}, R_{i2}, y R_{i3} son regiones resultado del proceso de segmentación previo, que en forma inherente (por pertenecer a la misma región de movimiento), tienen la capacidad de fusionarse mayormente. Por lo tanto, la probabilidad de ser seleccionadas como regiones pivote crece al aumentar su tamaño. El algoritmo trabaja sobre cuatro aspectos principales: Las regiones pueden seleccionarse como pivotes en más de una ocasión, la probabilidad de seleccionar regiones grandes como pivotes crece al aumentar su tamaño, al fusionar regiones, el parámetro descriptor de movimiento de la región pivote se adapta progresivamente al parámetro de la región global R_t, y finalmente, la región con mayor semejanza se fusiona a la región pivote.

La discusión anterior se limitó a explicar el fenómeno entre regiones conexas como R_{j1} y R_{j2}. No obstante es sencillo pronosticar que un fenómeno similar sucede entre regiones no conexas, como por ejemplo entre R_{j1} y R_{j3} de la figura 4.19.

Guía para la implementación en lenguaje C

El algoritmo anterior puede fácilmente transportarse al lenguaje de programación C, considerando la siguiente estructura y arreglos de memoria:

```
typedef struct {
  float Dx;
  float Dy;
  float Teta;
  float Kapa;
} TETA;                          /* VECTOR DE PARAMETROS */

unsigned int **ImReg;            /* IMAGEN DE REGIONES */
unsigned char **ImIn1;           /* IMAGEN EN T+1 */
unsigned char **ImIn;            /* IMAGEN EN T */
unsigned int VecEnReg[100]; /* VECINOS A UNA REGION */
TETA VecParam[100];
```

TETA se utiliza como un arreglo para almacenar los vectores de parámetros para cada región. Después del alojamiento en memoria, y lectura de archivos el algoritmo puede implementarse de la siguiente forma

```
REPITE
  Formar imagen de regiones
  Seleccionar un número de región aleatoria k
  Buscar regiones vecinas a k y almacenar en VecEnReg[Num]
  PARA l=0 HASTA Num
    Error_kl=0;
    Minimo=10000;
    PARA cada punto p que pertenece a la región k
      Error_kl=Error_kl+(It+1(p)-It(p+d(VecParam[l]))^2
    FIN PARA
    SI Error_kl<minimo entonces
      Minimo=Error_kl
      Region_a_fusionar=l
    Fin SI
  FIN PARA
  SI Minimo<umbral entonces fusiona Region_a_fusionar a k
HASTA el numero final de regiones
```

El algoritmo busca la región vecina con mayor semejanza a la región pivote k. Posteriormente si la región vecina cumple con el enunciado lógico para la fusión (ecuación 4.45), ésta se fusiona a la región pivote.

Algunas mejoras adicionales pueden implementarse. Por ejemplo se debe tener en cuenta que regiones de tamaño pequeño no pueden absorber regiones grandes. También el algoritmo puede finalizar cuando un cierto porcentaje de las regiones se ha cubierto.

Ejemplo de fusión de regiones

La figura 4.20 muestra el resultado final de la segmentación espacio-temporal, es decir, después de la fusión de regiones por movimiento.

Figura 4.20. *Miss América* 256X256X256 secuencias 38 y 40 después del filtrado, segmentación espacial y segmentación espacio-temporal. μ=20, regiones finales = 34.

Capitulo 5

Codificación

5.1. Introducción

Para la generación de algoritmos de codificación de video descrita en el presente trabajo, también conocida como orientada a regiones, la escena a codificar se segmenta en ciertas regiones, cada una de las cuales se codifica separadamente. En el caso ideal, las regiones identifican objetos. Cada región puede codificarse mediante dos conjuntos generales de parámetros descriptores: un conjunto para describir el movimiento de la región y el segundo que define la forma espacial de la región. En forma adicional, el enfoque de codificación de video presentado, contempla la transmisión de la primera secuencia del video codificada en JPEG, por ejemplo. Entonces el seguimiento espacial y temporal de las regiones independientes a partir de la primera secuencia proporciona la eliminación de redundancia, necesaria en la codificación a bajas tasas. Por el contrario, el proceso de reconstrucción toma información con baja redundancia de los parámetros descriptores de forma y movimiento además de la primera imagen en *to* para regenerar la información redundante que finalmente puede ser apreciada en forma agradable por el usuario.

El propósito general del presente capitulo es presentar las técnicas más comunes para la codificación de los contornos correspondientes a las regiones espacio-temporales y para la codificación de la imagen de error. La utilidad de la codificación está dirigida en dos aspectos importantes: por una parte, con respecto a las bajas tasas de bit que se pueden obtener en forma adicional a la eliminación de redundancia espacio-temporal; y por otra parte, con respecto a la mejora en la calidad de la imagen al compensar por errores introducidos en las etapas de estimación. El esquema de reconstrucción propuesto se muestra en la figura 5.1.

Entonces a partir del análisis de los procesos de codificación se realizan estimaciones de la tasa de compresión y de la calidad de la imagen reconstruida que se pueden obtener al utilizar el esquema de codificación de video propuesto (ver figuras 1 y 5.1).

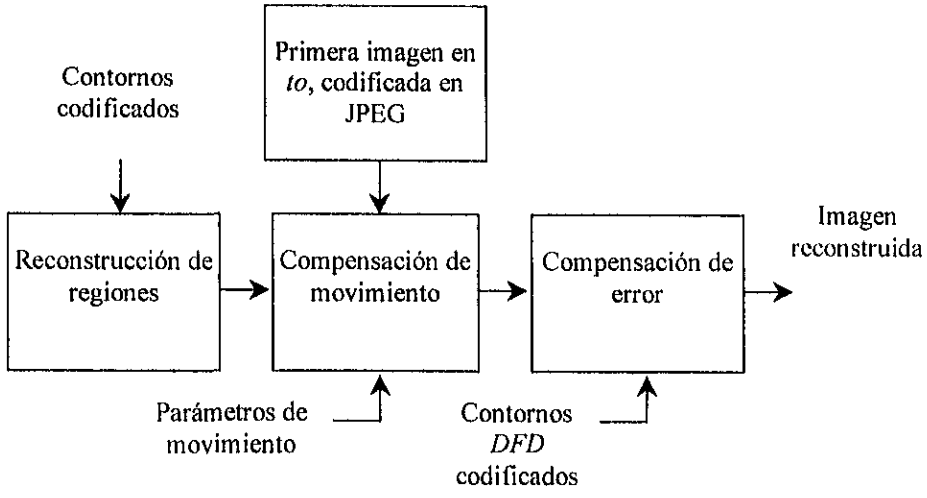


Figura 5.1. Reconstrucción de la secuencia de imágenes.

La segunda parte del capitulo presenta diversas técnicas para la codificación de la forma espacial de las regiones resultantes de la segmentación espacio-temporal. En la tercera parte planteamos un método para codificar el error de reconstrucción que puede utilizarse para mejorar la calidad de la imagen final.

5.2. Codificación de contornos

Las técnicas de segmentación espacio-temporal de los capítulos anteriores producen datos en crudo en forma de pixeles de un contorno o región. Aunque algunas veces estos datos se utilizan directamente para obtener descriptores (como para determinar la textura de una región), la práctica normalizada es utilizar esquemas que compacten los datos en representaciones que son considerablemente más útiles en el cálculo de los descriptores [3, 17,]. Adicionalmente se introduce mejora en la tasa de compresión, al eliminar redundancia en las cifras que representan datos.

Buscamos una representación que sea aplicada directamente a la representación en mapa de bits de la información de forma, ya que pasa por alto las complicaciones computacionales de una representación intermedia (Splines, polígonos, etc.) de contornos y sus conversiones asociadas.

5.2.1. Códigos de cadena

Los códigos de cadena se utilizan para representar un contorno por medio de una sucesión conexa de segmentos de longitud y longitud especificados. Normalmente esta representación se basa en segmentos de conectividad cuatro u ocho. La dirección de cada segmento se codifica utilizando un esquema de numeración como se indica en la figura 5.2.

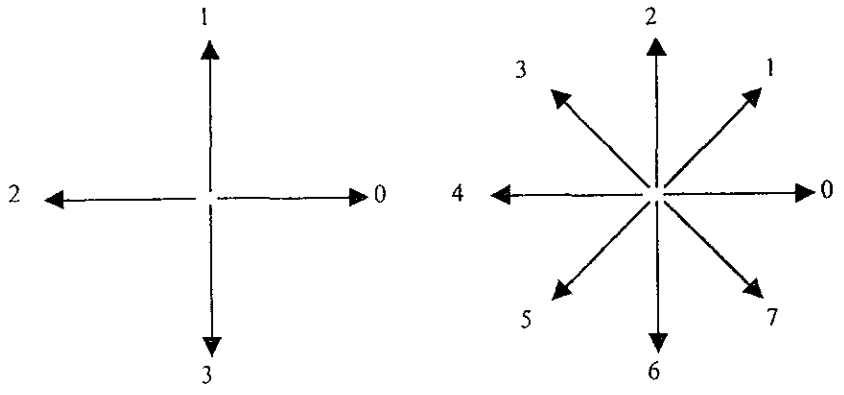


Figura 5.2. Direcciones de código, a). Código de cadena de 4 direcciones, b). Código de cadena de 8 direcciones.

Las imágenes digitales normalmente se adquieren y procesan en un formato de cuadriculas con igual espaciamiento en las direcciones x e y; por lo tanto, se puede generar un código de cadena siguiendo un contorno, por ejemplo, en el sentido de las manecillas del reloj y asignando una dirección a los segmentos que conectan cada par de pixeles. Este método generalmente es inaceptable por dos razones principales: 1) la cadena de códigos resultante es normalmente bastante larga, y 2) cualquier pequeña perturbación a lo largo del contorno debida al ruido o a una segmentación imperfecta origina cambios en el código, que pueden no estar necesariamente relacionados con la forma del contorno.

Una de las soluciones que se utiliza frecuentemente para soslayar los problemas mencionados consiste en volver a muestrear el contorno seleccionando un espacio de cuadrícula mayor, como se muestra en la figura 5.3. a). Entonces, según se recorre el contorno, se asigna un punto del mismo a cada nodo del cuadriculado, dependiendo de la proximidad del contorno original a ese nodo, como se muestra en la figura 5.3.b). El contorno remuestreado obtenido de esta forma se puede representar después por un código de 4 u 8, como se muestra en las figuras 5.3. c) y d) respectivamente. El punto de partida en la figura 5.3.c) está en el punto más grueso, y el contorno es el 4-camino externo más corto que permita el cuadriculado de la figura 5.3.b). La representación del contorno de la figura 5.3.c) es el código de cadena 003...01, y en la figura 5.3.d) es el código 076...12. Como cabría esperar, la precisión de la representación del código resultante depende del espaciado del cuadriculado de muestreo.

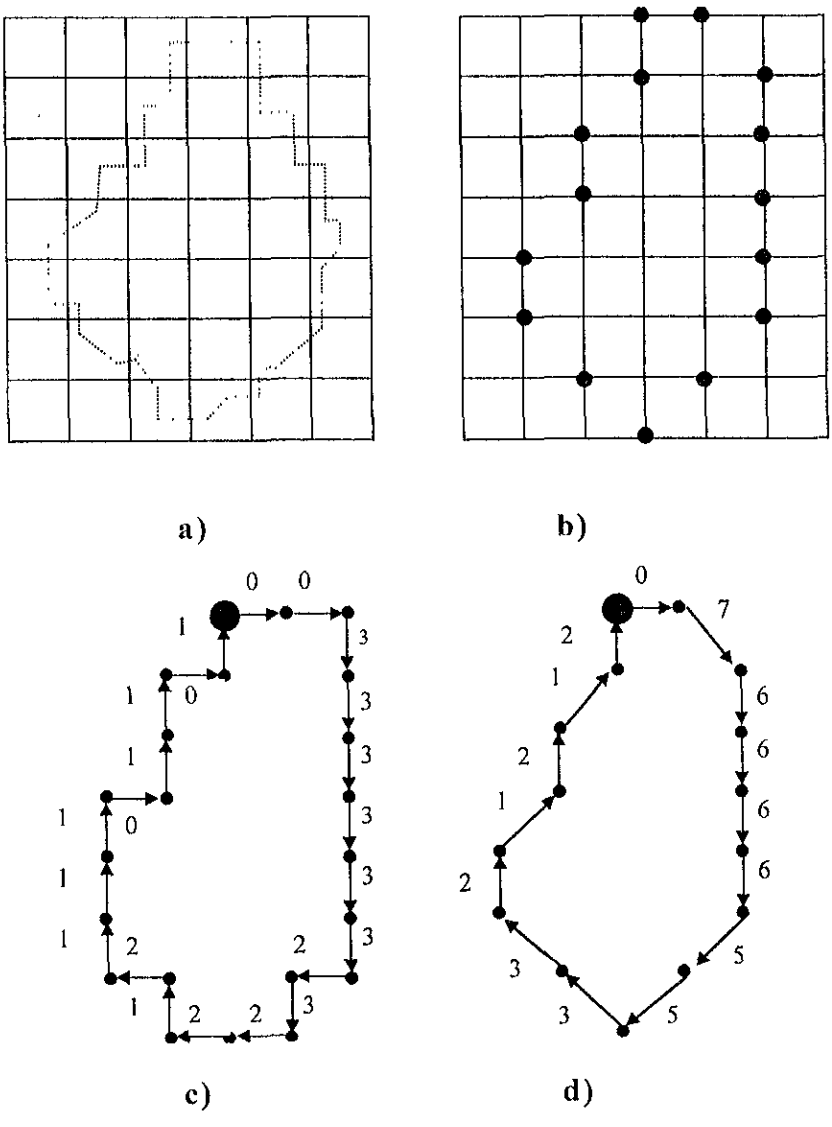


Figura 5.3. a). Contorno digital con el cuadriculado de remuestreo superpuesto, **b).** Resultado del muestreo, **c).** Código de cadena de 4 direcciones, **d).** Código de cadena de 8 direcciones.

El código de cadena de un contorno depende del punto de partida, pero dicho código se puede normalizar por un procedimiento directo: dado un código de cadena generado empezando en una posición arbitraria, se trata como una serie circular de números de dirección y se redefine el punto de partida de modo que la serie de números resultante forme un entero de módulo mínimo. También se puede normalizar por rotación utilizando la primera diferencia del código de cadena en vez del propio código. Esta diferencia se obtiene simplemente contando (en el sentido contrario a las manecillas del reloj) el número de direcciones que separan dos elementos adyacentes del código. Por ejemplo la primera diferencia del código de cadena de 4 direcciones 10103322 es 3133030. Si decidimos tratar el código como una serie circular, entonces el primer elemento de la diferencia se calcula

utilizando la transición entre los componentes último y primero de la cadena. Aquí el resultado es 33133030.

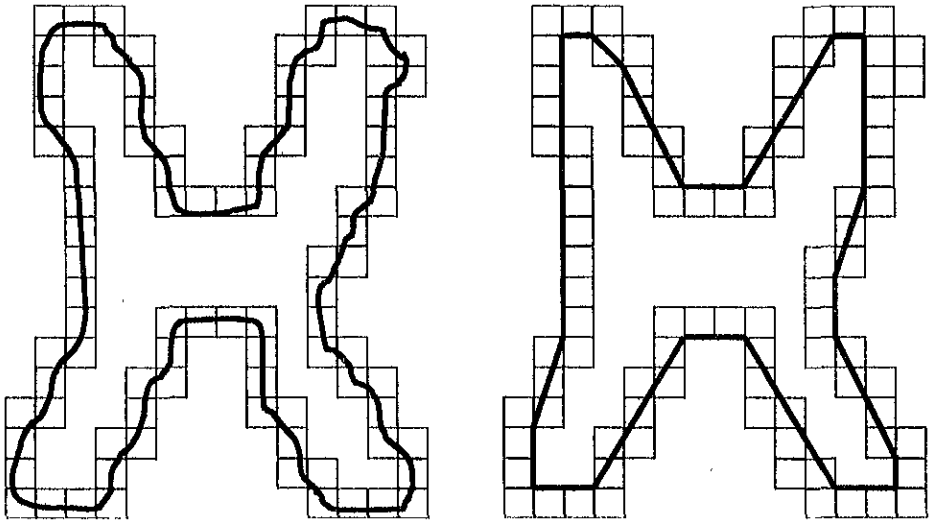
Estas normalizaciones son exactas solamente si los propios contornos son invariables a la rotación y al cambio de escala, lo cual, en la práctica, ocurre raramente. Por ejemplo, el mismo objeto digitalizado en dos orientaciones diferentes tendrá en general distintas formas de contorno con un grado de disimilitud proporcional a la resolución de la imagen. Este efecto se puede reducir seleccionando elementos de cadena que sean grandes en proporción con la distancia entre pixeles de la imagen digitalizada u orientado al cuadriculado de remuestreado a lo largo de los ejes principales del objeto a codificar.

Hasta ahora, el código de cadena parece ser la técnica más eficiente para la codificación de contornos. Al utilizar el concepto de 4-dirección, se ha demostrado que la técnica requiere en promedio 1.34 bits por punto del contorno [8]. Sin embargo, los contornos en un mapa de segmentación son muy largos. Estos ocupan una parte grande del costo total de codificación si se codifican directamente mediante el código de cadena.

5.2.2. Aproximaciones poligonales

Un contorno digital se puede tratar con una precisión arbitraria mediante un polígono. Para una curva cerrada, la aproximación es exacta cuando el número de lado del polígono es igual al número de puntos del contorno, de forma que cada par de puntos adyacentes define un lado del polígono. En la práctica, el objetivo de una aproximación poligonal es captar la esencia de la forma del contorno con un polígono con el menor número de lados posible. El problema en general no es trivial y se puede transformar rápidamente en una búsqueda iterativa de gran consumo de tiempo. Sin embargo, existen varias técnicas de aproximación poligonal de complejidad moderada y pequeñas necesidades de tratamiento, que son adecuadas para aplicaciones de procesamiento de imágenes.

En primer lugar, el procedimiento para encontrar polígonos de perímetro mínimo se explica mejor mediante un ejemplo. Supongamos que se encierra el contorno en un conjunto de células concatenadas, como se muestra en la figura 5.4.a). Ayudará a visualizar esta inclusión si se le observa como dos paredes correspondientes a los bordes exterior e interior de la sucesión de células y pensando que el contorno objeto es una tira de goma contenida entre las paredes. Si se permite que la tira de goma se encoja, tomará la forma de la figura 5.4.b), produciendo un polígono de perímetro mínimo que se adapta a la geometría establecida por la sucesión de células. Si cada célula abarca solamente un punto del contorno, el error en cada célula entre el contorno original y la aproximación de la tira de goma sería, como máximo, $(2d)^{1/2}$, siendo d la distancia entre pixeles. Este error se puede reducir a la mitad haciendo que cada célula este centrada en su pixel correspondiente.



a) b)

Figura 5.4. a). Contorno de un objeto encerrado por células, **b).** Polígono de perímetro mínimo.

Las técnicas de *fusión* basadas en errores o en otros criterios se han aplicado al problema de la aproximación poligonal. Una de las soluciones consiste en fusionar puntos a lo largo del contorno hasta que el ajuste de la curva de error mínimo cuadrado de los puntos fusionados, hasta el momento, traspase un umbral preestablecido. Cuando se cumple esta condición, se almacenan los perímetros de la curva, el error se pone a 0, y se repite el procedimiento fusionando nuevos puntos a lo largo del contorno hasta que el error, de nuevo, traspase el umbral. Al final del procedimiento las intersecciones de lados de líneas adyacentes forman los vértices del polígono. Una de las principales dificultades de este método es que los vértices generalmente no corresponden a inflexiones (tales como esquinas) del contorno, porque no se empieza una nueva línea hasta que el error no traspasa el umbral. Si, por ejemplo, se siguiera una larga línea recta y girara en una esquina, una vez pasada ésta se absorbería un cierto numero de puntos (dependiendo del umbral) antes de que se traspasara el umbral. Sin embargo, para aliviar esta dificultad se puede utilizar la división junto con la fusión.

Un método para *dividir* lados del contorno consiste en subdividir sucesivamente el lado en dos partes hasta que se satisfaga un criterio dado. Por ejemplo, un requisito podría ser que la distancia perpendicular máxima desde un lado del contorno a la línea que une sus dos extremos no exceda un umbral preestablecido Si lo hace, el punto más alejado se convierte en un vértice, subdividiendo así el lado en dos sublados. Esta aproximación tiene la ventaja de buscar puntos de inflexión destacados. Para un contorno cerrado, los mejores puntos para comenzar son normalmente los dos puntos mas separados del contorno. Por ejemplo, la figura 5.5.a) muestra el contorno de un objeto, y la figura 5.5.b) muestra una subdivisión de este contorno (línea continua) sobre sus puntos más separados. El punto *c* tiene la mayor distancia

perpendicular desde el lado superior a la línea *ab*. De la misma forma, el punto *d* tiene la mayor distancia en el lado inferior. La figura 5.5.c) muestra el resultado de utilizar el procedimiento de división con un umbral igual a 0.25 veces la longitud de la línea *ab*. Como ningún punto de los nuevos lados del contorno tiene una distancia perpendicular (a su recta correspondiente) que exceda este umbral, el procedimiento da como resultado el polígono de la figura 5.5.d).

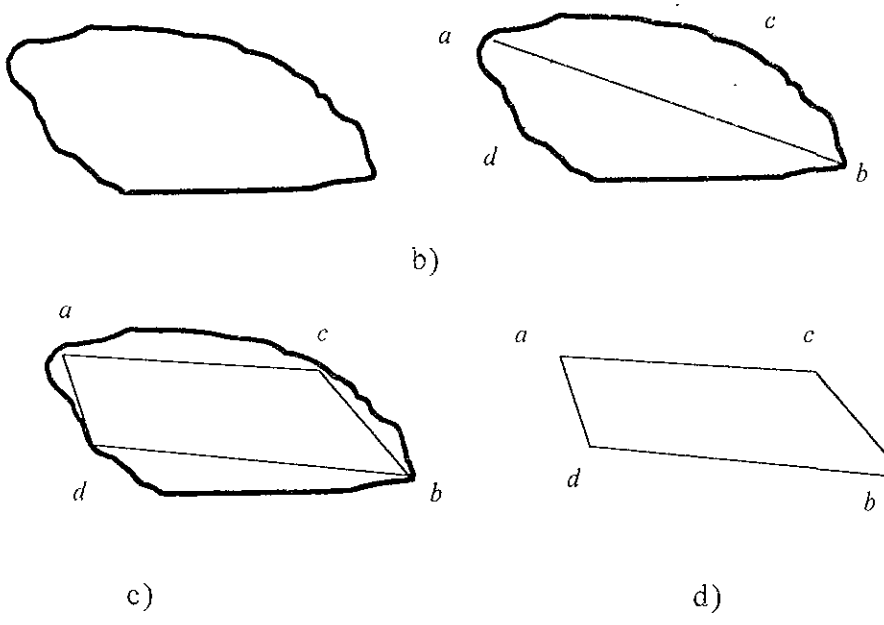


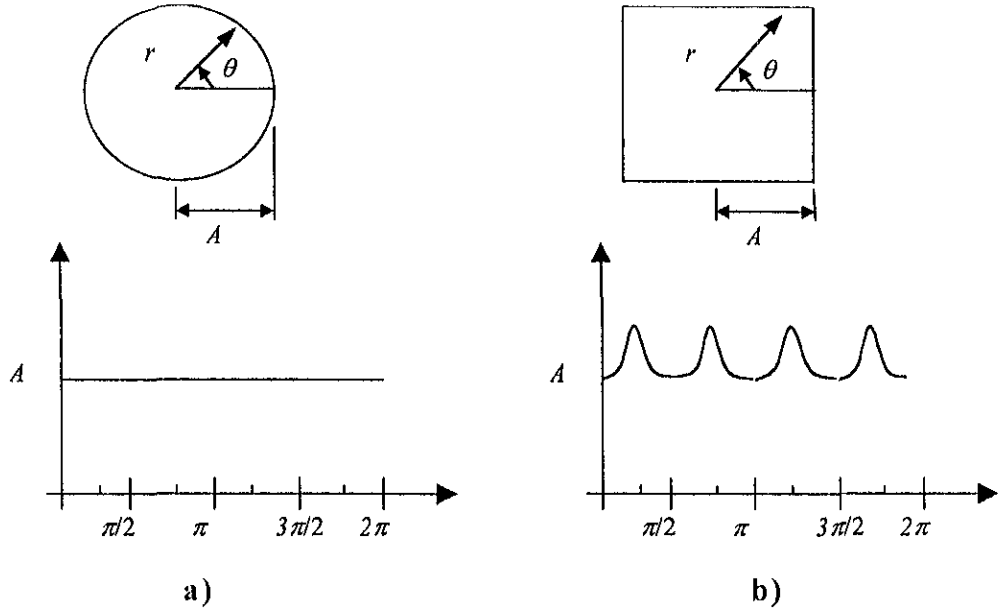
Figura 5.5. a). Contorno original, **b).** Contorno dividido en dos lados basándose en el cálculo de distancias. **c).** Unión de vértices, **d).** Polígono resultante.

5.2.3. Firmas

Una firma es una representación funcional unidimensional de un contorno y se puede generar de varias formas. Una de las más simples es representar la distancia desde el centro al contorno como una función del ángulo, como se ve en la figura 5.6. Sin embargo, independientemente de como se genere la firma, la idea básica es reducir la representación del contorno a una función unidimensional, que presumiblemente es más fácil de describir que el contorno original bidimensional.

Las firmas generadas por el procedimiento que se acaba de describir no varían con la traslación, pero dependen de la rotación y la escala. Se puede conseguir la normalización con respecto a la rotación encontrando un modo de seleccionar el mismo punto de partida para generar la firma, independientemente de la orientación de la forma. Un método para hacer esto consiste en seleccionar como punto de partida el punto más alejado del centro, si sucede que este punto es único e independiente de aberraciones rotacionales para cada forma de interés. Otro método consiste en seleccionar el punto del

eje propio principal (mayor) más alejado del centro. Este método requiere más cálculo pero es más consistente porque la dirección del eje principal se determina a partir de la matriz de covarianza, que se basa en todos los puntos de contorno. Otra forma consiste en obtener el código de cadena del contorno y utilizar después la solución de la sección 5.2.1, suponiendo que el código es lo suficientemente burdo para que la rotación no afecte a su circularidad.



a) b)

Figura 5.6. Dos curvas de contornos sencillos y sus correspondientes firmas de distancia-ángulo. **a)** $r(\theta)$ es constante, **b)** $r(\theta)$=A secθ.

Basándose en las suposiciones de uniformidad de escala con respecto a ambos ejes y que el muestreo se forma intervalos iguales de θ, los cambios de tamaño de una forma producen cambios en los valores de amplitud de la firma correspondiente. Un método sencillo de normalizar este resultado es escalar todas las funciones de tal manera que siempre abarquen el mismo rango de valores, por ejemplo [0, 1]. La principal ventaja de este método es la simplicidad, pero presenta la desventaja potencial de que el escalado de la función completa depende sólo de dos valores: el mínimo y el máximo. Si las formas tienen ruido esta dependencia puede ser una fuente de error de objeto a objeto. Un método más desigual (pero también más intensivo de calcular) consiste en dividir cada muestra por la varianza de la firma, suponiendo que dicha varianza no es cero -como en el caso de la figura 5.6.a)- o tan pequeña que cree dificultades de cálculo. El empleo de la varianza produce un factor de escala variable que es inversamente proporcional a los cambios de tamaño y funciona de forma muy parecida a un control automático de ganancia. Independientemente del método utilizado, la idea básica Consiste en eliminar la dependencia del tamaño pero conservar la forma fundamental de las formas de onda.

El método distancia-ángulo no es, desde luego, el único modo de generar una firma. Por ejemplo, se podría recorrer el contorno y dibujar el ángulo

entre una línea tangente al contorno y una línea de referencia como una función de posición a lo largo del contorno. La firma resultante, aunque bastante diferente de la curva $r(\theta)$, podría contener información acerca de las características básicas de la forma. Por ejemplo, los segmentos horizontales de la curva corresponderían a líneas rectas a lo largo del contorno, dado que el ángulo tangente seria constante ahí. Una variación de este método consiste en utilizar como firma la denominada *función de densidad de pendiente.* Esta función es simplemente un histograma de valores de ángulos tangentes. Como un histograma es una medida de concentración de valores, la función de densidad de pendiente responde con fuerza a secciones del contorno con ángulos tangentes constantes (lados rectos o casi rectos) y tiene profundos valles en las secciones que producen ángulos de variación rápida (esquinas u otras inflexiones agudas).

5.2.4. Lados del contorno

A veces es útil descomponer un contorno en lados. La descomposición reduce la complejidad del contorno y simplifica así el proceso de descripción. Esta solución es particularmente atractiva cuando el contorno presenta una o más concavidades significativas que contienen información sobre la forma. En este caso el empleo del cerco convexo de la región abarcada por el contorno es una poderosa herramienta para una descomposición robusta del contorno.

El *cerco convexo* H de un conjunto arbitrario S es el conjunto convexo más pequeño que contiene a S. El conjunto diferencia H-S se denomina *deficiencia convexa* D del conjunto S.

En la práctica los contornos digitales tienden a ser irregulares a causa de la digitalización, el ruido y las variaciones en la segmentación. Estos efectos normalmente producen una deficiencia convexa que tiene componentes pequeños, insignificantes, esparcidos aleatoriamente sobre el contorno. Mejor que intentar soslayar estas irregularidades en un proceso posterior, la práctica común consiste en suavizar el contorno antes de su división. Hay varios modos de hacerlo. Un método es recorrer el contorno y reemplazar las coordenadas de cada pixel por las coordenadas medias de m de sus vecinos a lo largo del contorno. Esta solución funciona para pequeñas irregularidades pero consume mucho tiempo y es difícil de controlar. Valores grandes de m pueden dar un excesivo suavizado, mientras que valores pequeños podrían no ser suficientes en algunos lados del contorno. Una técnica más desigual es utilizar una aproximación poligonal, como en la sección 5.2.2, antes de encontrar la deficiencia convexa de una región. Independientemente del método utilizado para el suavizado, la mayoría de los contornos digitales de interés son simples polígonos (polígonos sin autointersección).

Los conceptos del cerco convexo y su deficiencia son igualmente útiles para describir una región completa, así como su contorno. Por ejemplo, la descripción de una región se podría basar en su área y en la de su deficiencia

convexa, en el número de componentes de la deficiencia convexa, en la situación relativa de estos componentes, y así sucesivamente.

5.2.5. El esqueleto de una región

Una importante aproximación para representar la forma estructural de una región plana es reducirla a un gráfo. En esta reducción se puede conseguir el *esqueleto* de la región mediante un algoritmo de reducción (denominado también *esqueletización*). Los procedimientos de reducción tienen un papel primordial en una amplia gama de problemas del procesado de imágenes, abarcando desde la inspección automática de tarjetas de circuitos impresos hasta contar las fibras en los filtros de aire.

El esqueleto de una región se puede definir mediante la transformación del eje medio (MAT, del inglés *Medial Axis Transformation*). La MAT de una región R con borde B es la siguiente. Para cada punto P de R, se encuentra su vecino más próximo en B. Si P tiene más de un vecino de éstos, se dice que pertenece al eje medio (esqueleto) de R. El concepto de "más próximo" depende de la definición de una distancia, y por lo tanto los resultados de una operación MAT están influidos por la elección de una medida de distancia. La figura 5.8 muestra algunos ejemplos, que utilizan la distancia euclidiana.

Aunque la MAT de una región proporciona un esqueleto instintivamente atrayente, la implementación directa de esa definición normalmente es prohibitiva en términos de cálculo. La implementación implica potencialmente el cálculo de la distancia desde cada punto interior a cada punto del contorno de una región. Se han propuesto muchos algoritmos para mejorar la eficacia de cálculo, mientras se intenta a la vez producir una representación del eje medio de una región. Normalmente, estos son algoritmos de reducción que suprimen iterativamente los puntos del margen de una región sujetos a las restricciones que la supresión de estos puntos 1) no elimine puntos extremos, 2) no rompa la continuidad y 3) no cause excesiva erosión en la región.

En esta sección presentamos un algoritmo de reducción de regiones binarias. Se supone que los puntos de la región tienen valor 1 y los puntos del fondo tienen valor 0. El método consiste en pasadas sucesivas de dos pasos básicos aplicados a los puntos del contorno de una región dada, siendo un *punto de contorno* cualquier pixel con valor 1 que tenga al menos un 8-vecino con valor 0. Con referencia a la definición de 8-vecindad de la figura 5.9, el paso 1 marca un punto del contorno p para que sea eliminado si se satisfacen las siguientes condiciones:

a) $2 \leq N(p1) \leq 6$

b) $S(p1)=1$

c) $p2 \cdot p4 \cdot p6=0$

d) $p4 \cdot p6 \cdot p8=0$

En el paso 2, las condiciones a) y b) permanecen igual, pero las condiciones c) y d) se cambian a:

c') $p2 \cdot p4 \cdot p8=0$

d') $p2 \cdot p6 \cdot p8=0$

El paso 1 se aplica a todo pixel del borde de la región binaria considerada. Si se violan una o más de las condiciones c) y d), no se cambia el valor del punto en cuestión. Si se satisfacen todas las condiciones, se marca el punto para su supresión. Sin embargo, no se borra el punto hasta que todos los puntos del borde hayan sido procesados. Este retraso impide que se cambie la estructura de los datos durante la ejecución del algoritmo. Después de haber aplicado el paso 1 a todos los puntos del borde, se eliminan (se ponen a 0) aquellos que estaban marcados. A continuación se aplica el paso 2 a los datos resultantes exactamente de la misma forma que el paso 1.

De esta forma, una iteración del algoritmo de reducción consiste en: 1) aplicar el paso 1 para marcar los puntos del borde a suprimir; 2) borrar los puntos marcados; 3) aplicar el paso 2 para marcar los restantes puntos del borde para su eliminación; y 4) borrar los puntos marcados. Este procedimiento básico se aplica iterativamente hasta que no se suprimen más puntos, momento en el que termina el algoritmo, produciendo el esqueleto de la región.

La condición a) no se cumple cuando el punto $p1$ del borde tiene solamente uno o siete 8-vecinos con valor 1 El tener solamente uno de tales vecinos implica que el punto $p1$ es el extremo de un trazo del esqueleto y evidentemente no debería ser borrado. Borrar $p1$ si tiene 7 de estos vecinos causaría erosión en la región. La condición b) no se cumple cuando se aplica a puntos de un trazo de un pixel de grosor. Por tanto, esta condición evita la discontinuidad de segmentos de un esqueleto durante la operación de reducción. Las condiciones c) y d) se satisfacen simultáneamente por el conjunto mínimo de valores: ($p4=0$ o $p6=0$) o ($p2=0$ y $p8=0$). Por tanto, y con respecto a la estructura de vecindad de la figura 5.9, un punto que satisface estas condiciones, así como las a) y b), es un punto del borde sur o este o un punto de la esquina noroeste del contorno. En cualquier caso, $p1$ no forma parte del esqueleto y debería eliminarse. De igual forma, las condiciones c') y d') se satisfacen simultáneamente por el siguiente conjunto mínimo de valores ($p2=0$ o $p8=0$) o ($p4=0$ y $p6=0$). Estos corresponden a los puntos de los bordes norte u oeste, o al punto de la esquina sudeste. Obsérvese que los puntos de la esquina nordeste tienen $p2=0$ y $p4=0$ y por tanto satisfacen las condiciones c) y d) al igual que los c') y d'). Lo mismo es cierto para los puntos de la esquina sudoeste, que tienen $p6=0$ y $p8=0$.

Por otra parte, desde el punto de vista de representación morfológica, sea B un disco de radio r centrado en x. El esqueleto de X denotado como $SK(X)$, es el conjunto de discos máximos inscribibles en X:

$$SK(X)=\cup\{x\in X,\ \exists\ r>0 \mid B(x,r) \text{ es máximo dentro de } X\}$$

$$SK(X) = \bigcup_{r>0} Sr(X)$$

$$X = \bigcup_{r>0} B(Sr(X), r)$$

Como lo muestra la figura 5.7

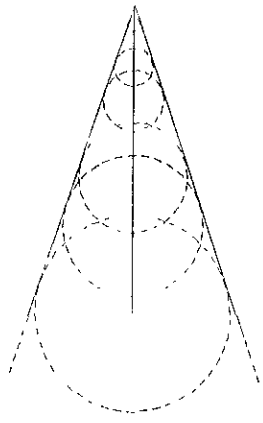


Figura 5.7. Representación morfológica del esqueleto mediante discos máximos.

En donde $Sr(X)$ es el subconjunto esqueleto de $SK(X)$ asociado al radio r. Adicionalmente, el esqueleto puede expresarse a partir de erosiones y aperturas:

$$SK(X) = \bigcup_{r>0} [(X \otimes rB) \setminus (X \otimes rB) \circ drB]$$

en donde rB corresponde al disco con radio r y drB representa un disco con radio infinitesimal dr. "\" es la diferencia.

También existe la transformación inversa:

$$X = SK^{-1}[SK(X)] = \bigcup_{r>0} [Sr(X) \oplus rB]$$

$$SK(X) = \bigcup_{n>0}^{N} Sn(X)$$

con $Sn(X)=[(X \otimes nB) \setminus (X \otimes nB) \circ B]$ y $N=\max\{n \mid X \otimes nB \neq \varnothing\}$

$Sn(X)$ corresponde al enésimo sub-esqueleto de X; si B es el elemento estructural, nB corresponde al elemento estructurante de tamaño n, obtenido al dilatar B n veces:

$$nB = B \oplus B \ldots \oplus B \ (n \text{ veces})$$

Para $n=0$, nB es definido como el origen (0, 0).

De otra forma:

$$X = \bigcup_{n=0}^{N} [Sn(X) \oplus nB]$$

El algoritmo puede considerarse como la descomposición a partir de detalles finos (n=0), hacia detalles burdos (n=N). Entonces es posible obtener una aproximación de la forma al omitir los índices menores. En éste sentido tenemos X en apertura por kB si sólo mantenemos índices mayores o iguales a k:

$$X \circ kB = \bigcup_{n=0}^{N} [Sn(X) \oplus nB]$$

La descomposición paso por paso se muestra en la figura 5.8.

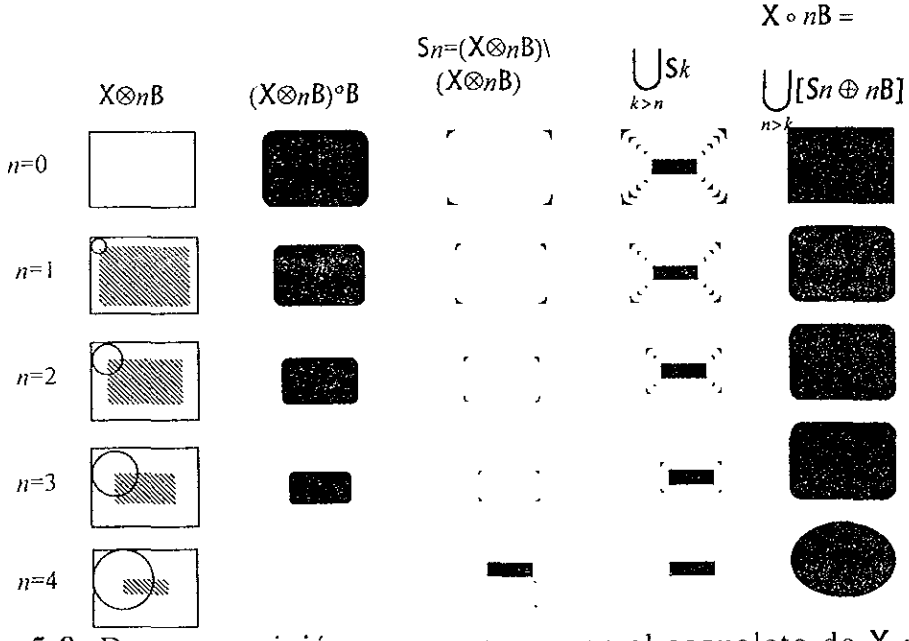


Figura 5.8. Descomposición paso por paso en el esqueleto de X por un elemento estructurante circular y la reconstrucción total o parcial de los indices siguientes.

La figura 5.9 muestra la aplicación del esqueleto morfológico sobre una imagen de franjas de interferencia óptica, originalmente en escala de gris.

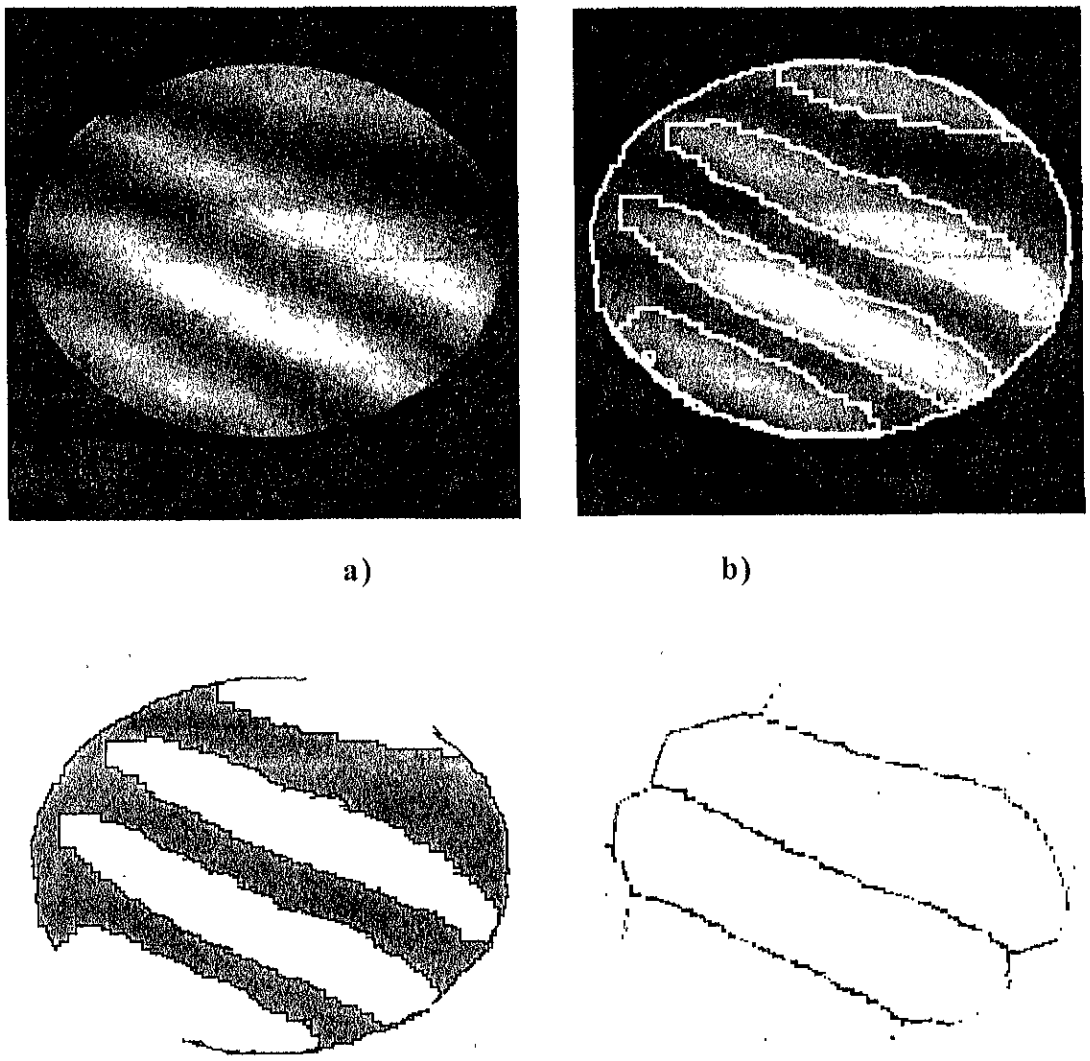


Figura 5.9. Imagen de franjas de interferencia, **a)** Imagen original, **b)** Imagen después del filtrado por reconstrucción y segmentación espacial, **c)** Regiones de interés aisladas, **d)** Esqueleto de regiones.

5.3. Codificación de la imagen de error

La compensación de errores que en forma inherente introduce el proceso de codificación utilizado, mejora notablemente la calidad de la imagen. Para generar la imagen de error, conocida como imagen *DFD*, en la parte codificadora se restan la imagen original en $t_0+\Delta t$ y la imagen reconstruida en $t_0+\Delta t$ al considerar segmentación espacio-temporal y estimación de

movimiento (ver figura 1). Posteriormente en la parte decodificadora se utiliza la imagen *DFD* para mejorar la calidad de la imagen reconstruida (ver figura 5.1).

En este sentido, comúnmente se adoptan dos enfoques para codificar la imagen *DFD*: 1) Codificación orientada a pixeles, 2) codificación orientada a regiones. En la primera, la técnica más empleada es la Transformada Coseno Discreta (*Discrete Cosine Transform, DCT*). En la segunda, la técnica puede ser la descrita en este trabajo, utilizando segmentación espacial y codificación de contornos. En general, una imagen *DFD* consiste de algunas estructuras lineales y un número importante de pequeñas manchas que aparecen en el fondo con valores cercanos a cero. La figura 5.10 muestra la imagen *DFD* de la secuencia 40 de *Miss América* con aumento de contraste.

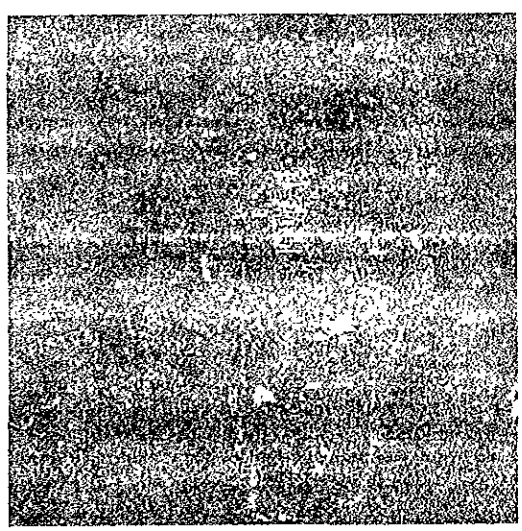


Figura 5.10. Imagen *DFD* de *Miss América.*

5.3.1. Codificación de imágenes usando la Transformada Coseno Discreta (DCT)

Probablemente el uso más común de las transformadas en dos dimensiones es la compresión de imágenes. Muchas transformadas han sido implementadas en compresión y su desempeño puede ser comparado por la fidelidad de la imagen recuperada con la original. El flujo general de los datos de la imagen en compresión por transformación es el mostrado en las figuras 5.11 y 5.12. La imagen muestreada es transformada. Un número de coeficientes resultantes son elegidos del total y los restantes coeficientes son re-cuantizados con menor número de bits que los originales. Para recuperar la imagen el proceso opuesto es realizado: los coeficientes son re-cuantizados hacia el número original de bits, los coeficientes desconocidos son reemplazados por valores fijos y la transformación inversa se calcula.

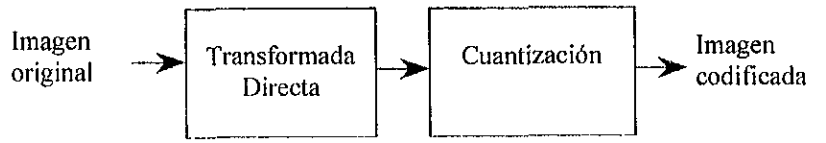


Figura 5.11. Compresión por transformación.

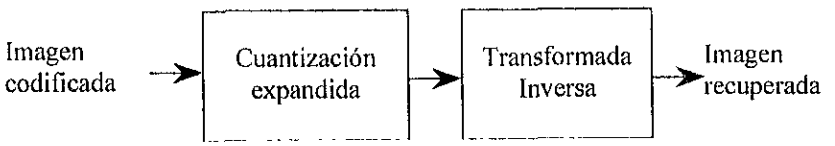


Figura 5.12. Recuperación de una imagen comprimida.

La fidelidad del proceso puede medirse al tomar la diferencia en nivel de intensidad entre las imágenes original y recuperada en cada punto del arreglo. Si la original es $I(\mathbf{p})$ y la recuperada es $I'(\mathbf{p})$ (ambos arreglos de NXN) entonces la diferencia del arreglo es

$$I_{dif}(\mathbf{p}) = I(\mathbf{p}) - 'I(\mathbf{p}) \tag{5.1}$$

El índice *PSNR* mide la fidelidad que se puede obtener del arreglo de diferencias.

Ya que la DCT proporciona error cuadrático medio (MSE) cercano al límite teórico, la DCT es muy popular en compresión de imágenes. La definición de la DCT para una imagen de NXN es

$$F(u,v) = \frac{4c(u)c(v)}{n^2}\sum_{j=0}^{n-1}\sum_{k=0}^{n-1} f(j,k)\cos\left[\frac{(2j+1)u\pi}{2n}\right]\cos\left[\frac{(2k+1)v\pi}{2n}\right] \tag{5.2}$$

en donde

u, v = variables discretas en la frecuencia (0, 1, 2, ..., N-1)
$f(j,k)$ = N X N pixeles de imagen (0, 1, 2, ..., N-1)
$F(u, v)$ = resultado de la DCT

$$c(w) = \begin{cases} 1 \\ \sqrt{2} \end{cases} w = 0$$
$$1 w = 1,2,...n-1$$

La DCT inversa está definida como

$$f(j,k) = \sum_{u=0}^{n-1}\sum_{v=0}^{n-1} c(u)c(v)F(u,v)\cos\left[\frac{(2j+1)u\pi}{2n}\right]\cos\left[\frac{(2k+1)v\pi}{2n}\right] \tag{5.3}$$

en donde

j, k = índices de la imagen resultante (0, 1, 2, ..., N-1)
$F(u, v)$ = resultado de la DCT a ser transformado inversamente de N X N
$f(j,k)$ = N X N pixeles de imagen (0, 1, 2, ..., N-1)

Cuantización de los coeficientes en la imagen comprimida

La razón por la cual la imagen puede comprimirse por factores grandes y recuperarse exitosamente con errores pequeños es la gran cantidad de redundancia en imágenes típicas. Claramente si un arreglo de números tiene redundancia es teóricamente posible proporcionar la misma información con menos números. El propósito de desarrollar la transformada es el desarrollo de un conjunto de números que representen la imagen pero cuyos valores estén descorrelacionados. Ya que el mismo contenido de la información será representado en los arreglos original y transformado, algunos números en el arreglo transformado proporcionan poca o ninguna información acerca de la imagen original y pueden ser descartados.

La verdadera medición de lo innecesario de un coeficiente dado es su varianza sobre un conjunto de imágenes. Si un coeficiente mantiene el mismo valor sobre el mismo conjunto de imágenes, entonces éste no proporciona mucha información y puede ser reemplazado por una constante en el extremo receptor sin dañar la fidelidad. Contrariamente, si un coeficiente tiene alta varianza sobre el conjunto, entonces no puede ser descartado sin consecuencias serias para la imagen recuperada.

En la DCT, las varianzas sobre conjuntos típicos tienden a tener contornos con varianza constante como se muestra en la figura 5.13. Los coeficientes con más alta varianza tienden a estar cercanos al origen de los ejes u y v. Una manera de usar ésta característica es asignar un número de niveles de cuantización basados en la Barinas de los coeficientes. Esto es conocido como codificación por zona. Un segmento de éste tipo puede reducir el número de bits requeridos para representar una imagen de 8 bits por pixel a 1.5 bits por pixel.

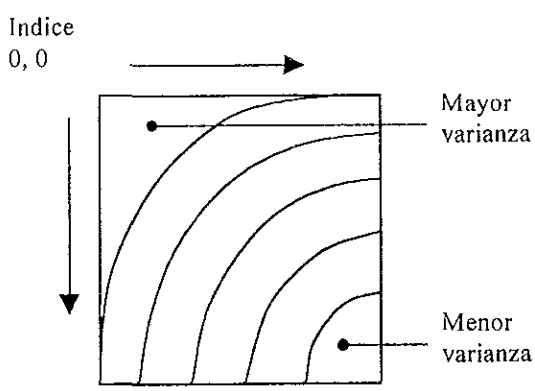


Figura 5.13. Lineas de variación de coeficientes típicas en una imagen transformada.

Codificación de bloque

En la codificación de bloque se utiliza una partición similar a la empleada en el algoritmo de la Transformada Rápida de Fourier (*Fast Fourier Transform, FFT*) para reducir dramáticamente el tiempo de procesamiento. La

técnica divide una imagen en muchas sub-imágenes. Una transformada entonces se desarrolla sobre cada una de las sub-imágenes y los coeficientes son cuantizados justo como si cada uno perteneciera a diferentes imágenes. La tabla 5.1 muestra una comparación de las operaciones de la transformación para una codificación de imagen completa contra la codificación de bloque para diversos tamaños de bloque.

Tamaño de bloque	Factor en el tiempo de transformación	Número requerido	Total	Factor de acelera-miento
64X64	$17X10^6$	16	$3X10^8$	16
32X32	$1X10^4$	64	$64X10^6$	64
16X16	65536	256	$16X10^6$	256
8X8	4096	1024	$4X10^6$	1024
4X4	256	4096	$1X10^6$	4096

Tabla 5.1. Comparación en los tiempos de transformación para una imagen completa (256X256) y codificación de bloque.

5.3.2. Codificación de la imagen *DFD* usando la segmentación espacial

Un enfoque que mejora el desempeño en la codificación de la imagen de error es considerar sólo las zonas espaciales que contienen información. Al contrario de la codificación por DCT, en la cual se da el mismo peso a cualquier zona de la imagen, la segmentación elimina la redundancia espacial en aquellas zonas amplias que introducen gran redundancia en la codificación. En el caso particular de la imagen *DFD* de secuencias típicas de videoconferencia o videoteléfono, la imagen contiene pequeñas manchas aisladas pero concentradas, correspondientes a las zonas en donde la compensación de movimiento deja un error remanente.

Para mejorar las relaciones de compresión, el tamaño pequeño y la baja energía de las manchas debe ser codificado en un número reducido de datos, como lo expuesto en [29]. Sea $D(\mathbf{p})$ una imagen *DFD*. Las manchas de baja energía son eliminadas al umbralizar $D(\mathbf{p})$ mediante el valor T:

$$d_1(\mathbf{p}) = \begin{cases} 1, & D(\mathbf{p}) > T \\ 0, & \text{otro caso} \end{cases} \qquad d_2(\mathbf{p}) = \begin{cases} 1, & D(\mathbf{p}) < -T \\ 0, & \text{otro caso} \end{cases}$$

Entonces las manchas pequeñas son removidas mediante filtros morfológicos por reconstrucción multi-direccionales.

Por ejemplo, para $d_1(\mathbf{p})$ tenemos:

$$d_3(\mathbf{p}) = \bigvee_{i=4}^{4} \gamma_{B_i}[d_1(\mathbf{p})] \qquad d_4(\mathbf{p}) = \gamma^{(\text{rec})}[d_3(\mathbf{p}), d_1(\mathbf{p})]$$

en donde *Bi* son los elementos estructurantes de longitud 5 y orientación 0°, 45°, 90° y 135°, respectivamente. No obstante, pueden existir pequeños hoyos en $d_4(\mathbf{p})$. Para eliminar estos hoyos, $d_4(\mathbf{p})$ es filtrada mediante el cierre con un elemento estructurante de 3X3. El mismo proceso se aplica a $d_2(\mathbf{p})$. Entonces los dos resultados se combinan en un solo mapa de segmentación. Cada región en el mapa de segmentación se aproxima por su nivel medio de gris. Esto constituye la primera reconstrucción de $D(\mathbf{p})$. Posteriormente, la imagen de error de reconstrucción se segmenta iterativamente mediante el método descrito y el mapa de segmentación resultante es superpuesto en el primer mapa de segmentación hasta que no se obtengan nuevas regiones. Este proceso iterativo garantiza que no hay manchas ni estructuras de tamaño grande. Por lo tanto el proceso de reconstrucción tiene un desempeño aceptable.

De igual forma, los contornos en el mapa de segmentación se codifican utilizando el código de cadena descrito en la sección anterior. Los niveles de gris promedios se pueden codificar con un número menor de bits, por ejemplo 6. La imagen *DFD* de la figura 5.10 es segmentada en 30 regiones espaciales. Se puede demostrar que el proceso de segmentación tiene buen desempeño. La figura 5.14 muestra la imagen *DFD* codificada utilizando la técnica descrita. El proceso ilustrado extrae toda la energía importante de las regiones en la imagen *DFD*. El costo de la codificación es aproximadamente 0.0061 bits por pixel, mientras que el *PSNR* se incrementa en aproximadamente 1 dB.

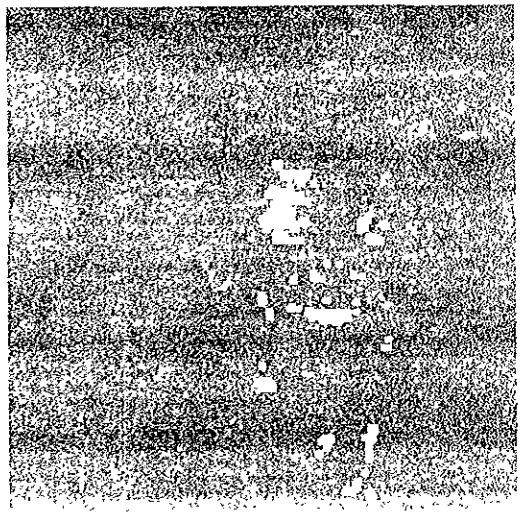


Figura 5.14. Imagen *DFD* de la figura 5.10 codificada.

Algunas mejoras adicionales pueden introducirse como parte de la codificación de la imagen *DFD*. Por ejemplo, este procedimiento iterativo puede resultar en un número grande de regiones, especialmente regiones pequeñas. En primer lugar, cada región menor a un porcentaje del total del área de la imagen puede fusionarse a la región adyacente con mayor similitud. Entonces, dos regiones adyacentes son fusionadas en una región si la diferencia entre sus niveles promedio de gris es menor que un umbral T o si el número de regiones es mayor que un número N predeterminado.

Capitulo 6

Resultados y conclusiones

6.1. Introducción

Finalmente, resumimos el trabajo realizado con la presentación de conclusiones, enfatizando en el trabajo a futuro. En los resultados, se muestran los dos parámetros de mayor importancia (tasa de compresión y calidad de la imagen) que proporcionan una visión general acerca de los resultados obtenidos que fácilmente pueden compararse con los trabajos de otros autores.

6.2. Resultados

El desempeño de los algoritmos presentados se puede resumir mediante dos gráficas que expresan la calidad de las imágenes y la tasa de compresión como función del número de cuadro en la secuencia de video. Es obvio que la calidad de la imagen va en decremento conforme el número de cuadro aumenta. La razón es que nuevas regiones aparecen y desaparecen. Otra razón es que la restricción de conservación en la iluminación pierde fuerza conforme avanzan los cuadros. Inclusive contemplamos la penalización del algoritmo cuando la calidad de la imagen reconstruida cae debajo de un cierto umbral.

La calidad de la imagen reconstruida se estima de la siguiente forma:

$$DFD^2 = \frac{1}{N}\sum_{\mathbf{p}\in I}[I_t(\mathbf{p}) - I(I_{t+1}(\mathbf{p}+\mathbf{d})]^2 \qquad (6.1)$$

en donde $I_t(\mathbf{p})$ es la imagen original e $I_t' = I(I_{t+1}(\mathbf{p}+\mathbf{d}))$ es la imagen reconstruida después de la segmentación espacio temporal y estimación de movimiento. Sobre la base del *DFD* se calcula el parámetro más común que proporciona una medida casi estándar en procesamiento de imágenes para estimar la calidad de una imagen:

$$PSNR = 20\log_{10}\left(\frac{255}{DFD}\right) \tag{6.2}$$

En la situación en que el *PSNR* indica una mala calidad en la imagen, se renueva la segmentación espacio temporal para posteriormente sólo relizar la estimación de movimiento. El algoritmo simple utilizado se muestra en la figura 6.1 y complementa el esquema de compresión planteado en la figura 1.

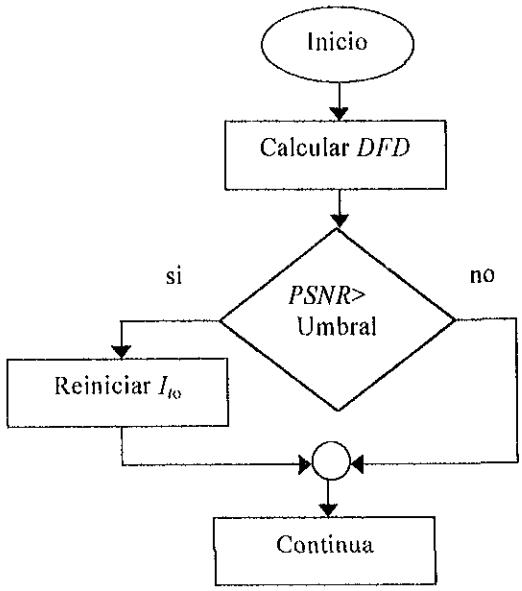


Figura 6.1. Algoritmo para el control de la calidad en la imagen en las secuencias de video.

Por otra parte la cuantificación de la relación de compresión se realiza sobre la base de los siguientes parámetros que permiten reconstruir la imagen, según el esquema de reconstrucción planteado en la figura 5.1.

En la tabla 6.1, la primera imagen se codifica en JPEG, para obtener una relación de compresión típica de 4 a1. En la codificación de contornos, el número de bytes asignados se calcula dinámicamente, ya que el patrón de intensidades, y por lo tanto la longitud y complejidad de los contornos, influye en tal cantidad. Los parámetros de movimiento, fijados a 4 como parte del modelo de movimiento, resultan ser 4 por región. En éste sentido, el número de regiones puede ser controlado en la segmentación espacio-temporal y se puede determinar de forma experimental. Por otra parte, para la imagen de error se procede de igual forma, sólo que ahora disminuye enormemente la primera imagen codificada en JPEG y el número de contornos codificados. Lo anterior nos da la idea de que entre mejor sea la estimación de movimiento, menor será la redundancia temporal y la imagen de error tiende a cero.

Parámetro	Cantidad	Bytes asignados
1ª Imagen	256X256	16384
Contornos codificados	Depende de la imagen y número de regiones	Se asignan dinámicamente
Parámetros de movimiento	4	4 por región

Tabla 6.1. Parámetros para la reconstrucción de la imagen.

Como se mencionó, la variación en la intensidad luminosa influye grandemente en la renovación de la segmentación espacio-temporal. De esta forma, para escenas que cambian constantemente de iluminación o aprecen y desaparecen regiones, la cantidad de parámetros para reconstruir crece de forma importante. Lo anterior no sucede para el caso de escenas restringidas; tal es el caso de videoteléfono o videoconferencia.

Las gráficas de compresión y calidad de la imagen (medida en *Peak Signal to Noise Ratio PSNR*, Relación Señal a Ruido Pico) que se obtienen para la secuencia *Miss América* se muestran en las figuras 6.2 y 6.3.

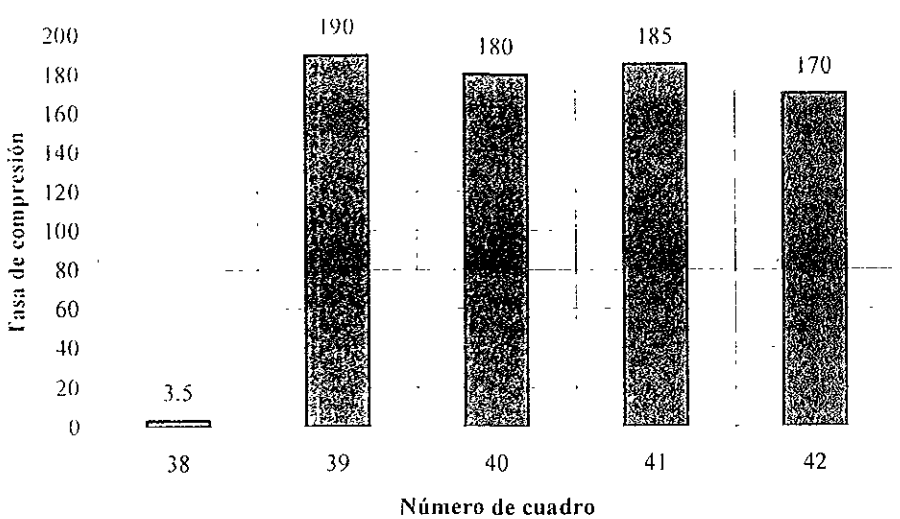


Figura 6.2. Tasa de compresión contra número de cuadro en la secuencia *Miss América*.

La gráfica en la figura 6.3 muestra como la primera tasa de compresión es extremadamente baja (similar a JPEG). No obstante, conforme avanzan los cuadros, avanza la estimación de movimiento y disminuye el número de parámetros a transmitir. Lo anterior proporciona números elevados en la tasa de compresión. Si las regiones en la escena permanecieran constantes, entonces el algoritmo no necesitaría penalización por pérdida en la calidad de la imagen, lo cual proporcionaría muy altas tasas de compresión para un número grande de cuadros.

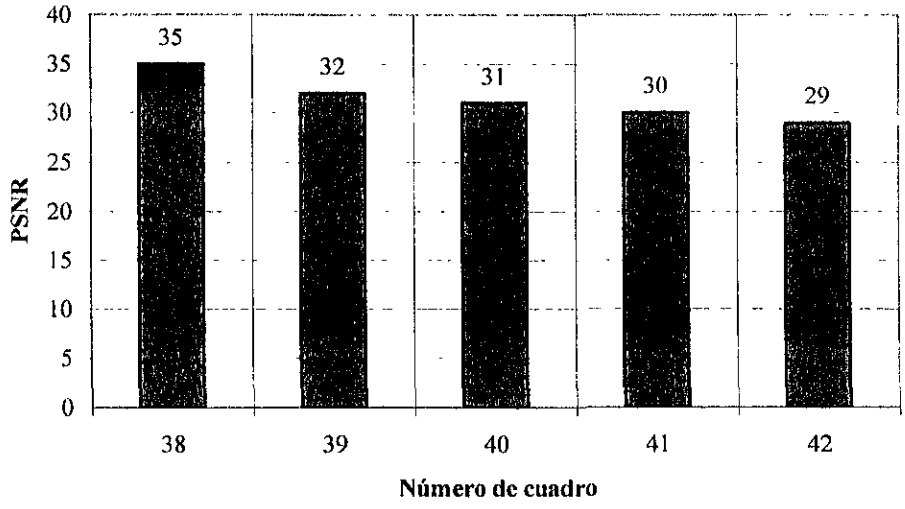


Figura 6.3. *PSNR* contra número de cuadro en la secuencia *Miss América*.

La gráfica en la figura 6.3 muestra claramente como la calidad en la imagen se deteriora conforme avanzan los cuadros. La tendencia es que para un número grande de cuadros, el algoritmo penalice por baja calidad de la imagen reconstruida, dando origen a una nueva segmentación espacio-temporal de refresco. Tal refresco en la segmentación contribuye nuevamente a una tasa de compresión baja, como se muestra para el cuadro inicial de la figura 6.3.

6.3. Conclusiones y trabajo a futuro

Se ha presentado un trabajo para la codificación de video orientado a regiones. En lugar del tradicional enfoque orientado a pixeles (DCT), el enfoque a regiones ofrece notables ventajas en cuanto a la baja tasa de transmisión que se obtiene. No obstante la complejidad de los algoritmos presentados, las crecientes capacidades de cómputo permiten la implementación en aplicaciones que no impliquen la operación en tiempo real. Inclusive, el enfoque geométrico y tratamiento aritmético presentados, permiten la implementación en arquitecturas paralelo y operaciones alambradas que permitan acelerar el proceso. Otra técnica que se puede utilizar para acelerar el tiempo de procesamiento, puede ser similar a la usada para implementar la Transformada Rápida de Fourier (*Fast Fourier Transform, FFT*), en donde se realizan divisiones elementales del conjunto global de datos.

Por otra parte, el enfoque orientado a objetos permite la representación jerárquica de ciertos "objetos visuales" (en forma general y en terminología técnica MPEG-4, *Audio/Visual Objects, AVO's*), como por ejemplo [10]:

a) Un escenario de fondo 2D

b) La extracción de objetos de interés sin el fondo

Tal representación jerárquica permite construir escenas complejas y habilita a los consumidores a manipular objetos individuales.

En un nivel de procesamiento superior, el enfoque orientado a regiones permitirá representar objetos cumpliendo algunos requerimientos básicos para el video MPEG-4 como:

a) Poner objetos en cualquier lugar dentro de un sistema de ejes coordenados.

b) Agrupar objetos primitivos con el objeto de crear nuevos ambientes.

c) Caracterizar objetos mediante parámetros que permitan modificar sus atributos (por ejemplo, cambiar texturas, animar rostros).

d) Cambiar en forma interactiva el sistema de visión 3D.

e) Arrastrar objetos en la escena original hacia posiciones diferentes.

f) Construir "flujos" de datos pertenecientes a objetos que puedan ser multiplexados por los medios de comunicación limitados en ancho de banda, administrando eficientemente los recursos.

g) Tasas de bit entre 5 y 64 kbits/s.

h) 15 cuadros por segundo.

i) Resolución de 352X288 pixeles para la iluminancia y 176X144 pixeles para la crominancia (*Common Intermediate Format, CIF*).

j) Interacción con el usuario. Por ejemplo, modificar los atributos de una escena, hacer visibles o invisibles algunos objetos, introducir nuevos objetos ya sea naturales o sintéticos, etc.

k) Introducción de objetos visuales que identifiquen derechos de autor o diversas publicidades de terceros.

Por otra parte y con respecto al enfoque adoptado en este trabajo, las etapas de segmentación espacial y estimación de movimiento se basan fuertemente en una técnica orientada a regiones. Nosotros enfrentamos los dos problemas anteriores orientando las técnicas hacia la compresión de video. No obstante, es bien conocido que tales técnicas constituyen en la actualidad un conjunto de herramientas utilizadas ampliamente en otras aplicaciones de visión por computadora. Un ejemplo es visión estéreo, en donde se pueden implementar algoritmos de segmentación y estimación de movimiento cooperativos para encontrar la correspondencia entre pares de puntos, conocida como *stereo matching*. A continuación presentamos un resumen de los principales esfuerzos enfocados hacia la segmentación espacial y estimación de movimiento, así como examinamos algunas cuestiones abiertas y trabajo a futuro.

6.3.1. Segmentación espacial

Este es uno de los problemas más importantes en procesamiento de imágenes [7]. El problema de segmentación espacial puede ser visto como la búsqueda de una forma de subdividir el dominio de una imagen en regiones que representen la proyección de partes visibles pertenecientes a objetos en la escena. Si bien la tarea es sencilla para los seres humanos y algunos seres vivos, su implementación eficiente sobre un sistema de cómputo no es trivial.

Con respecto al esfuerzo realizado en torno a la segmentación espacial, este fue dividido en cinco etapas principales: simplificación, división, fusión, eliminación de regiones pequeñas y control del número final de regiones:

1. Para la etapa de simplificación, el operador *opening-closing* por reconstrucción es propuesto como herramienta para eliminar detalles menos relevantes perceptualmente sin corromper los bordes de la imagen. El objetivo es reducir la carga computacional y remarcar las decisiones de las etapas posteriores.
2. En la etapa de división en regiones por *árbol cuádruple*, la imagen es sobre-segmentada, al utilizar un criterio de homogeneidad, en un esquema coherente con el patrón de intensidades sobre la imagen. El objetivo es proporcionar un nivel de procesamiento basado en regiones en lugar de nivel pixel.
3. La etapa de fusión de regiones produce una sobre segmentación notablemente menor que en la etapa anterior, al utilizar fusión de regiones adyacentes que cumplen con un criterio de similitud. El objetivo es disminuir la sobre-segmentación y progresar en forma significativa hacia el aspecto visual.
4. La etapa de eliminación de pequeñas regiones fusiona pequeñas regiones contrastantes, resultantes del paso previo, con sus vecinos más dominantes perceptualmente.
5. El control en el número de regiones permite orientar la segmentación espacial, si se evalúan con conocimiento a priori ciertas características de la escena. Por ejemplo, las escenas típicas de videteléfono o videoconferencia se pueden restringir a ambientes con fondo estático y una persona mostrando cabeza y hombros. En éste sentido, el número de regiones se puede limitar para proporcionar tasas de bit apropiadas al medio de comunicación.

Los procesos aleatorios involucrados en la segmentación espacial, implican un tiempo de procesamiento impredecible. No obstante, con las técnicas modernas de cómputo, el algoritmo se adapta bien al procesamiento paralelo y a las arquitecturas hardware, debido a la subdivisión en etapas y a su orientación geométrica.

Los umbrales de decisión alteran de manera importante el desempeño de los algoritmos; la adaptabilidad a diferentes escenarios es una área de trabajo que

debe mejorarse con procesamiento previo que estime umbrales óptimos bajo diversas circunstancias.

6.3.2. Estimación de movimiento

En cuanto a la estimación de movimiento, no obstante que se comprobó que la calidad de la imagen reconstruida aparentemente es superior para el caso del *estimador cuadrático*, la *coherencia de los parámetros de movimiento* es otro punto que debe tomarse en cuenta. En éste sentido, existen dos argumentos que justifican el uso del enfoque robusto. El primero es que un número reducido de parámetros es suficiente para describir completamente los vectores de parámetros, que pueden ser grandes, y tales vectores constituir una aproximación más coherente al flujo óptico real. El segundo argumento es el costo en tiempo de cálculo.

En compresión de video, la estimación de movimiento se convierte en una clara evidencia y necesidad para eliminar redundancia temporal a partir de la recuperación del flujo óptico. No obstante la importancia de la estimación de movimiento, ésta es sólo una herramienta en visión por computadora.

En este caso, el problema es enfrentado como el análisis de un conjunto de imágenes estáticas. Para sistemas de visión dinámicos, la estimación de movimiento constituye una fuente de información abundante que permite interactuar con el mundo cambiante.

La idea básica detrás del trabajo presentado, es formular el problema de estimación de movimiento como una función objetivo a minimizar, con restricciones espaciales y temporales apropiadas. Para tal fin, existe un gran número de técnicas numéricas que se pueden emplear. Algunas adoptan enfoques estocásticos, otros se basan en el principio del gradiente, y en otros casos en donde se presentan movimientos grandes, el problema es jerarquizado en forma piramidal desde una aproximación burda hasta un cálculo fino.

Algunos aspectos de interés acerca de la estimación de movimiento son los siguientes:

a) La estimación robusta de movimiento tiene como objetivo fortalecer la inmunidad a gradientes espaciales localizados en pequeñas áreas y, de esta manera, fortalecer la coherencia temporal.

b) Por lo tanto, el enfoque robusto se plantea como una alternativa de lisado en el campo de movimiento que mejora el desempeño del enfoque cuadrático.

c) Se plantea el problema de estimación de movimiento básicamente como un problema de minimización.

d) Se plantea una solución analítica para la minimización sobre la base del método del gradiente, en donde las derivadas espaciales juegan un papel importante.

e) La técnica iterativa de minimización utilizada converge rápidamente (cinco iteraciones).

f) No obstante la suposición de intensidad constante, la técnica planteada funciona aceptablemente para el caso de aplicaciones restringidas (por ejemplo, videoteléfono, videoconferencia) en donde la variación de iluminación en el ambiente no es muy significativa.

6.3.3. Preguntas abiertas y trabajo a futuro

El trabajo presentado aquí nos proporciona la pauta para trabajar a futuro en los siguientes aspectos de gran interés:

- Buscar métodos para representar los contornos de las regiones en formas más suaves, agradables para la vista. Técnicas basadas en crecimiento o disminución de regiones a nivel pixel o representación matemática (por ejemplo, curvas de Bezier o splines).
- Discutir criterios automáticos eficientes para la división, fusión y eliminación con mayor relación al conocimiento previo de la imagen o a objetivos deseables, para diferentes imágenes.
- Discutir criterios alternativos para la fusión y eliminación que tomen en consideración texturas.
- Investigación de técnicas de codificación de regiones orientadas a la transmisión de vídeo.
- Procesamiento posterior para la representación y descripción de regiones.
- Estudiar criterios de segmentación espacial que busquen por dos condiciones funcionales: alcanzar un número mínimo de regiones uniformes, y que el tamaño de las regiones sea los más grande posible.
- Plantear una función de minimización de energía para la segmentación espacial, que mida la calidad de lisado en las curvas de segmentación.
- Introducir modelos de movimiento que contemplen la variación en la intensidad de la iluminación.
- Introducir técnicas de minimización estocásticas e inmunes a los gradientes espaciales muy localizados.
- Introducir modelos de movimiento basados en superficies 3D de orden superior para el análisis de estructuras deformables.
- Introducir visión estéreo para la elaboración de modelos de movimiento más robustos, menos ambiguos, y posiblemente visión dinámica.
- Introducir algoritmos cooperativos entre la segmentación espacial y la estimación de movimiento.
- Cuantificación del beneficio/costo de diferentes algoritmos de segmentación espacio-temporal.

- Cuantificación de la incertidumbre en la estimación de movimiento.
- Procesamiento de áreas ocultas y áreas de reciente aparición que puedan afectar la calidad en la imagen.
- Estudiar criterios para la cuantificación de la ganancia en la tasa de bit potencial que aporta cada etapa en el proceso de codificación.
- Sobre la base de la medida anterior, optimizar las etapas que potencialmente afecten el desempeño de la tasa de bit.
- Extracción de características relevantes con aplicación *a metrología dimensional* [1]. Por ejemplo, detección de geometrías típicas como esquinas, bordes, marcas de trazadores láser, etc.

Anexo A

Minimización basada en el gradiente

A.1. Introducción

La forma más sencilla para minimizar una función objetivo $f(u_1,\ldots, u_n)$ de varias incógnitas es calcular sus derivadas parciales con respecto a cada incógnita, poner éstas a cero y resolver las ecuaciones resultantes [4, 18]:

$$\frac{\partial f(\mathbf{u})}{\partial u_1} = 0$$

$$\frac{\partial f(\mathbf{u})}{\partial u_n} = 0 \qquad \text{(A.1)}$$

simultáneamente para $u_1,\ldots, u_n$. Este conjunto de ecuaciones simultaneas pueden expresarse como una ecuación vectorial:

$$\nabla_u f(\mathbf{u})=0 \qquad \text{(A.2)}$$

en donde ∇_u es el operador gradiente con respecto al vector incógnita **u**. Ya que es difícil definir una función criterio de forma cerrada $f(u_1,\ldots, u_n)$ para la estimación de movimiento, y/o resolver el conjunto de ecuaciones A.2 en forma cerrada se puede recurrir a métodos numéricos. Por ejemplo, el *DFD* es una función de intensidades de pixeles que no puede expresarse en forma cerrada.

A.2. Método de descenso por pasos

El descenso por pasos es probablemente el método más sencillo de optimización numérica. Este renu8eva el estimador actual de la localidad del mínimo en la dirección del gradiente navegante, llamada la dirección del descenso por pasos. Recordar que el vector gradiente apunta en la dirección del máximo. Es decir, en una dimensión (función de una variable sencilla), su signo será positivo en una pendiente hacia arriba. Por lo tanto, la dirección

del paso descendente esta justamente en la dirección opuesta, lo cual se muestra en la figura A.1.

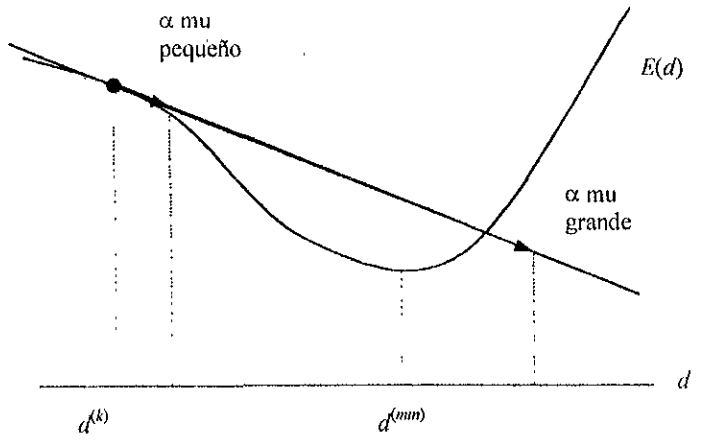


Figura A.1. Ilustración del método descendiente utilizando el gradiente.

En el sentido de obtener el más cercano al mínimo, se renueva el estimador actual de la siguiente forma:

$$\mathbf{u}^{(k+1)}=\mathbf{u}^{(k)}-\alpha\nabla_{\mathbf{u}}f(\mathbf{u})\,|\,\mathbf{u}^{(k)} \qquad (A.3)$$

en donde α es algún escalar positivo conocido como el tamaño de paso. El tamaño del paso es critico para la convergencia de las iteraciones, ya que si α es muy pequeño, entonces se desplaza en pequeños incrementos cada vez, y las iteraciones tomaran mucho tiempo en converger al mínimo. Por otra parte, si α es muy grande el algoritmo se vuelve inestable y tenderá a oscilar alrededor del mínimo. En el método del descenso por pasos, el tamaño de paso usualmente se elige heurísticamente.

A.3. Método de Newton-Raphson

El valor óptimo para el tamaño de paso α se puede estimar usando el método bien conocido de Newton-Raphson para encontrar raíces. Aquí por simplicidad, se muestra la derivación para el caso de un una función de una variable sencilla. En una dimensión, se desea hallar una raíz de $f'(u)$. Para este efecto, se expande $f'(u)$ en serie de Taylor en el punto $u^{(k)}$ para obtener:

$$f'(u^{(k+1)})=f'(u^{(k)})+(u^{(k+1)}-u^{(k)})f''(u^{(k)}) \qquad (A.4)$$

Ya que deseamos $u^{(k+1)}$ sea un cero de $f'(u)$, se especifica

$$f'(u^{(k)})+(u^{(k+1)}-u^{(k)})f''(u^{(k)})=0 \qquad (A.5)$$

Resolviendo la ecuación A.5 para $u^{(k+1)}$, tenemos:

$$u^{(k+1)} = u^{(k)} - \frac{f'(u^{(k)})}{f''(u^{(k)})} \tag{A.6}$$

Este resultado puede generalizarse al caso de una función de varias incógnitas de la siguiente forma:

$$\mathbf{u}^{(k+1)}=\mathbf{u}^{(k)}-\mathbf{H}^{-1}\nabla_{\mathbf{u}}f(\mathbf{u})\,|\,\mathbf{u}^{(k)} \tag{A.7}$$

en donde **H** es la matriz Hessiana.

A.4. Mínimos locales vs globales

El enfoque de descenso por pasos sufre de ciertas desventajas: la solución depende del valor inicial. Si se comienza en un valle la búsqueda caerá en el fondo de ese valle, aún si es un mínimo local. Ya que el vector gradiente es cero o cercano a cero, alrededor del mínimo local, las recursiones se vuelven pequeñas para el método que se mueve fuera del mínimo local. Una solución a este problema es iniciar el algoritmo en varios puntos iniciales, y entonces seleccionar la solución que proporciona el valor mínimo de la función de criterio.

Algunos métodos más complejos, tal como la alineación simulada, se encuentran documentados ampliamente en la literatura. Estos encuentran mínimos globales sin importar el punto inicial. Sin embargo, tales métodos consumen demasiado tiempo de cálculo.

Anexo B

Visión estéreo

B.1. Introducción

Actualmente existe un esfuerzo importante por aplicar técnicas de visión por computadora a la metrología dimensional [1, 9]. Sin embargo, para mediciones geométricas con exactitud, el problema principal está relacionado con los fenómenos de distorsión en la imagen plana [26, 27, 30]. Es bien conocido que algunas de las características que deben ser mejoradas en las cámaras y procedimientos de adquisición digital de imágenes son las siguientes: 1) Resolución espacial de la imagen, 2) Distorsión en los lentes, 3) Alineamientos mecánicos en la construcción de las cámaras, 4) Técnicas de reconocimiento de patrones y visión por computadora. Los primeros tres puntos están ligados fuertemente a los avances tecnológicos en los procesos de manufactura. Por otra parte, el punto cuatro además de abordar el problema particular a cada aplicación, adopta el enfoque de corrección por programación para enfrentar en forma elegante los primeros tres puntos.

Un nivel primario de visión por computadora útil en metrología dimensional, es sólo proporcionar *herramientas auxiliares para la visión*; es decir, imágenes de referencia que sobrepuestas al objeto bajo inspección ocular proporcionen información cualitativa. Adicionalmente, las técnicas de filtrado espacial, y mejora de imágenes pueden acentuar o eliminar, respectivamente, detalles importantes o ruido en las imágenes que puedan carecer de importancia perceptual.

En un nivel intermedio, la medición de características geométricas en imágenes binarias puede ser de utilidad en la automatización de mediciones monoculares. En éste sentido la *reconstrucción de los datos* toma una serie de mediciones geométricas planas bajo condiciones restringidas para estimar su situación en el espacio tridimensional.

En un nivel avanzado, en lo correspondiente al área de *visión estéreo con exactitud geométrica*, al problema se le ha dado un enfoque de programación, como parte de los procesos de *calibración*, *reconstrucción* y *apareamiento estéreo*. En visión estéreo, la *calibración* comprende el establecimiento de los parámetros geométricos y de corrección que permiten modelar un ambiente

global 3D a partir de sus proyecciones planas conocidas como *par estéreo*. Por otro lado, la *reconstrucción* comprende la generación del ambiente global 3D a partir del modelo y de pares de imágenes [24]. El *apareamiento estéreo* es una herramienta auxiliar de procesamiento de imágenes utilizada en los procesos de calibración y reconstrucción para la correlación de pares de puntos o regiones de las imágenes.

Esta sección plantea una aplicación de *visión estéreo* para determinar la situación espacial de los objetos, como una alternativa al empleo de Máquinas de Medir en Coordenadas, en aplicaciones que por la exactitud requerida así lo permitan.

B.2. Visión estéreo

La visión estéreo habilita a los seres humanos y sistemas de visión a interactuar poderosamente con el ambiente 3D que los rodea. El método para reconstruir un objeto o posición tridimensional a partir de dos de sus proyecciones planas mitiga la ambigüedad de escala (entre profundidad y translación) inherentes en la visión monocular, ya que un par de imágenes apropiadamente registradas contienen información acerca de la estructura de la escena.

La actividad de interés entonces resulta ser la extracción de objetos para estimar su situación en el espacio, imitando de alguna manera el complejo sistema de visión humano. La técnica de visión estéreo ha demostrado exactitud y economía en aplicaciones médicas, artísticas y de nivel industrial.

En resumen, la teoría de visión estéreo moderna se apoya y fundamenta en áreas de desarrollo particulares como:

- Calibración de cámaras. Encontrar el modelo matemático que relacione el arreglo físico y el par estéreo con las coordenadas globales 3D.
- Reconstrucción 3D a partir de un par estéreo. Reconstruir el ambiente 3D a partir del modelo anterior y un par estéreo.
- Apareamiento estéreo (stereo matching). Encontrar la correspondencia de regiones 2D en el par estéreo, en relación con las coordenadas 3D para su empleo en la calibración y reconstrucción.

B.3. Modelo de cámara

El sistema de visión estéreo se plantea sobre la base del modelo de la figura B.1 [26]. El modelo contempla los parámetros geométricos, de distorsión y de digitalización de imágenes involucrados. El modelo contempla las siguientes conversiones:

a) Convertir de coordenadas globales $[x_W\ y_W\ z_W]^{\mathbf{T}}$ a coordenadas de cámara $[x_C\ y_C\ z_C]^{\mathbf{T}}$:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} r1 & r2 & r3 \\ r4 & r5 & r6 \\ r7 & r8 & r9 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \tag{B.1}$$

en donde

T_X, T_Y, T_Z : Traslación para la transformación de coordenadas [mm].

R_X, R_Y, R_Z : Rotación para la transformación de coordenadas [rad].

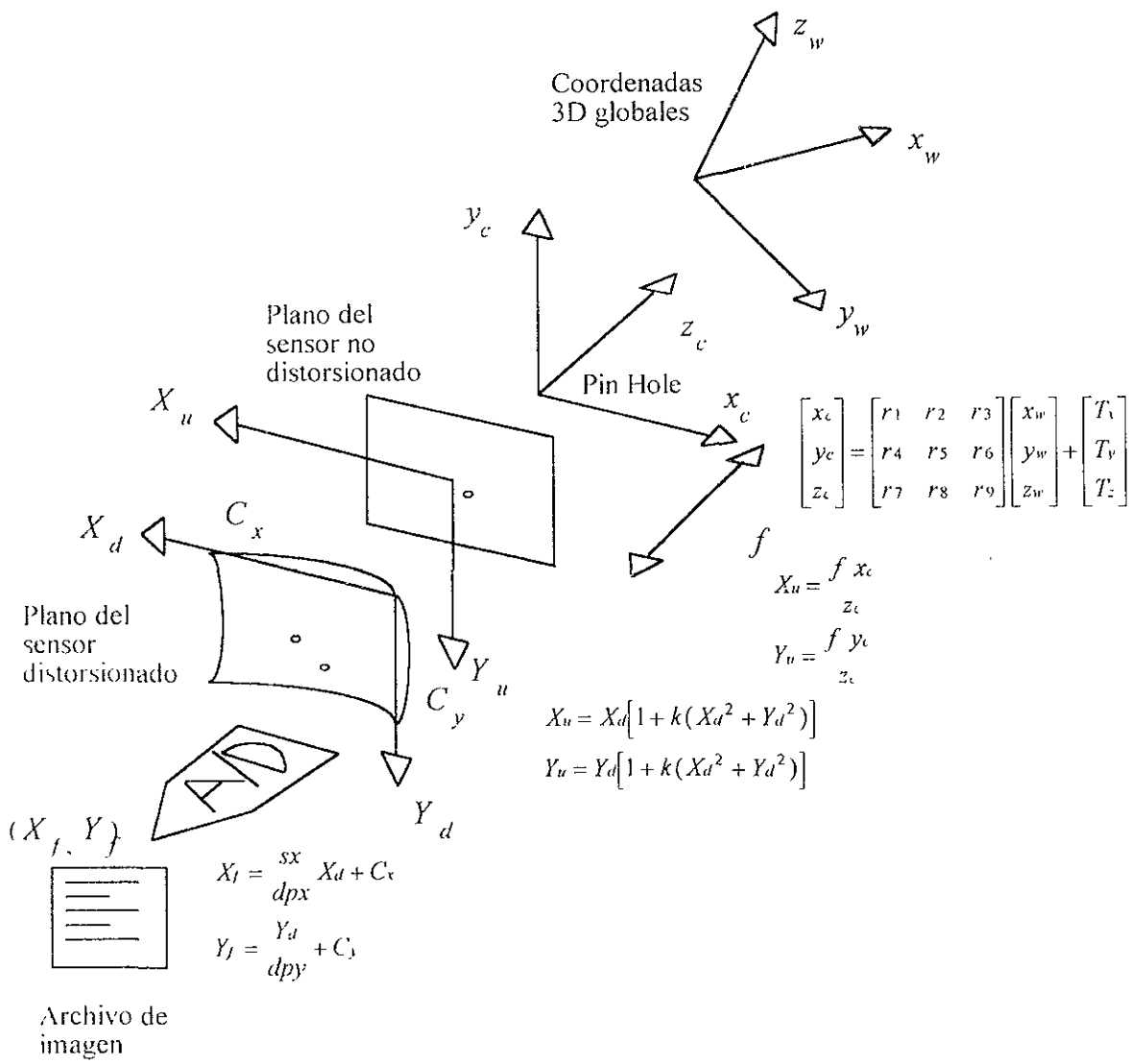


Figura B.1. Modelo de cámara pin hole propuesto por Tsai.

además

$r_1 = \cos(R_y) * \cos(R_z)$

$r_2 = \cos(R_z) * \text{sen}(R_x) * \text{sen}(R_y) - \cos(R_x) * \text{sen}(R_z)$

$r_3 = \text{sen}(R_x) * \text{sen}(R_z) + \cos(R_x) * \cos(R_z) * \text{sen}(R_y)$

$r_4 = \cos(R_y) * \text{sen}(R_z)$

$r_5 = \text{sen}(R_x) * \text{sen}(R_y) * \text{sen}(R_z) + \cos(R_x) * \cos(R_z)$

$r_6 = \cos(R_x) * \text{sen}(R_y) * \text{sen}(R_z) - \cos(R_z) * \text{sen}(R_x)$

$r_7 = - \text{sen}(R_y)$

$r_8 = \cos(R_y) * \text{sen}(R_x)$

$r_9 = \cos(R_x) * \cos(R_y)$

b) Convertir de coordenadas de cámara $[x_c\ y_c\ z_c]^{\mathbf{T}}$ a coordenadas de sensor no distorsionado

$$[X_u\ Y_u]^{\mathbf{T}}:$$

$$X_u = \frac{f x_c}{z_c}$$
$$Y_u = \frac{f y_c}{z_c} \qquad \text{(B.2)}$$

en donde f es la longitud focal de la cámara pin hole [mm].

c) Convertir de coordenadas de sensor no distorsionado $[X_u\ Y_u]^{\mathbf{T}}$ a coordenadas de sensor distorsionado $[X_d\ Y_d]^{\mathbf{T}}$

$$X_u = X_d\left[1 + k(X_d^2 + X_d^2)\right]$$
$$Y_u = Y_d\left[1 + k(X_d^2 + X_d^2)\right] \qquad \text{(B.3)}$$

en donde k es el coeficiente de distorsión de primer orden para un lente radial [1/mm^2].

d) Convertir de coordenadas de sensor distorsionado $[X_d\ Y_d]^{\mathbf{T}}$ a coordenadas de imagen $[X_f\ Y_f]^{\mathbf{T}}$:

$$X_f = \frac{sx}{dpx} X_d + C_x$$
$$Y_f = \frac{Y_d}{dpy} + C_y \qquad \text{(B.4)}$$

en donde

sx : Factor de escala para compensar incertidumbres en el rastreo horizontal del digitalizador.

dpx : Dimensión efectiva en x del pixel en el digitalizador [mm/pixel].

dpy : Dimensión efectiva en y del pixel en el digitalizador [mm/pixel].

Cx, Cy : Coordenadas del centro radial del lente [pixel].

B.4. Calibracion de cámaras

Tiene como objetivo estimar los parámetros del modelo anterior sobre la base del conocimiento de puntos 3D en el sistema de coordenadas global y sus proyecciones en el plano 2D. El modelo anterior se puede plantear como un sistema de ecuaciones

$$[X_f, Y_f] = \mathbf{F}\{[x_w, y_w, z_w][R_x, R_y, R_z, T_x, T_y, T_z, f, k, C_x, C_y, sx, dpx, dpy]\} \quad \text{(B.5)}$$

En donde se indica explícitamente en la ecuación **F** la relación entre las variables dependientes $[X_f, Y_f]$ conocidas, las variables independientes $[x_w, y_w, z_w]$ conocidas y los parámetros $[R_x, R_y, R_z, T_x, T_y, T_z, f, k, C_x, C_y, sx, dpx, dpy]$ desconocidos a estimar.

El método aplicado para estimar los parámetros puede ser Levenberg-Marquardt, ya que es bastante popular y está cobrando importancia como método estándar en el modelado de sistemas no lineales [4, 18].

B.5. Reconstrucción 3D a partir de un par estéreo

Tiene como objetivo la reconstrucción de coordenadas 3D de un punto **P** a partir del conocimiento de dos de sus proyecciones 2D derecha (X_f^D , Y_f^D) e izquierda (X_f^I , Y_f^I) y de los parámetros del modelo estimados como parte de la calibración (figura B.2). De las ecuaciones (B.1) y (B.2)

$$z_c \frac{X_u}{f} = r_1 x_w + r_2 y_w + r_3 z_w + T_x \quad \text{(B.6)}$$

$$z_c \frac{Y_u}{f} = r_4 x_w + r_5 y_w + r_6 z_w + T_y \quad \text{(B.7)}$$

$$z_c = r_7 x_w + r_8 y_w + r_9 z_w + T_z \quad \text{(B.8)}$$

sustituyendo (B.8) en (B.6) y (B.7)

$$(r_7x_w + r_8y_w + r_9z_w + T_z)\frac{X_u}{f} = r_1x_w + r_2y_w + r_3z_w + T_x \tag{B.9}$$

$$(r_7x_w + r_8y_w + r_9z_w + T_z)\frac{Y_u}{f} = r_4x_w + r_5y_w + r_6z_w + T_y \tag{B.10}$$

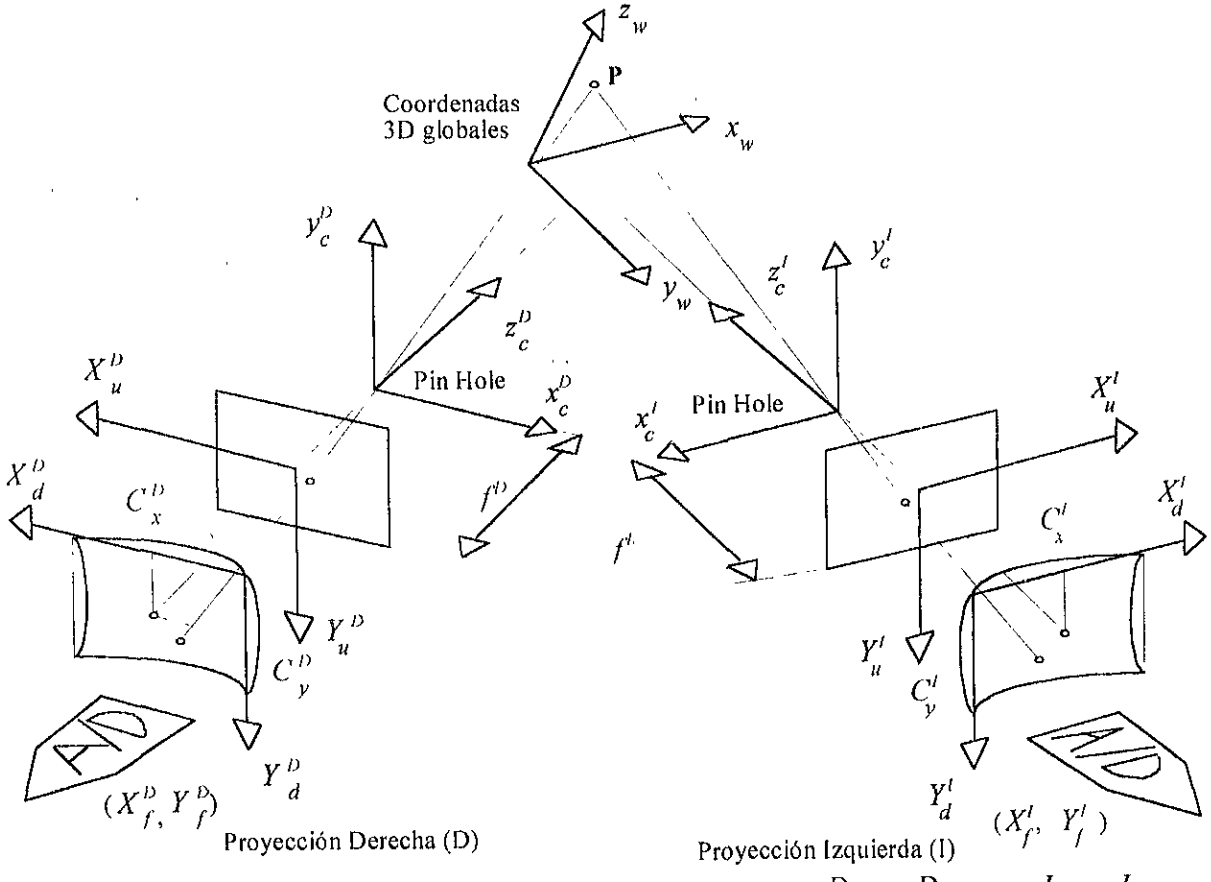


Figura B.2. Reconstrucción de **P** a partir de $(X_f{}^D\ ,\ Y_f{}^D)$ y $(X_f{}^I\ ,\ Y_f{}^I)$.

agrupando

$$(r_7\frac{X_u}{f} - r_1)x_w + (r_8\frac{X_u}{f} - r_2)y_w + (r_9\frac{X_u}{f} - r_3)z_w + (T_z\frac{X_u}{f} - T_x) = 0 \tag{B.11}$$

$$(r_7\frac{Y_u}{f} - r_4)x_w + (r_8\frac{Y_u}{f} - r_5)y_w + (r_9\frac{Y_u}{f} - r_6)z_w + (T_z\frac{Y_u}{f} - T_y) = 0 \tag{B.12}$$

(B.11) y (B.12) forman un sistema singular de dos ecuaciones con tres incógnitas. Para formar un sistema de ecuaciones no singular se toma un par de proyecciones derecha (D) e izquierda (I) del mismo punto en coordenadas globales

$$(r_7{}^D\frac{X_u{}^D}{f^D} - r_1{}^D)x_w + (r_8{}^D\frac{X_u{}^D}{f^D} - r_2{}^D)y_w + (r_9{}^D\frac{X_u{}^D}{f^D} - r_3{}^D)z_w + (T_z{}^D\frac{X_u{}^D}{f^D} - T_x{}^D) = 0 \tag{B.13}$$

$$(r_7^I \frac{X_u^I}{f^I} - r_1^I)x_w + (r_8^I \frac{X_u^I}{f^I} - r_2^I)y_w + (r_9^I \frac{X_u^I}{f^I} - r_3^I)z_w + (T_z^I \frac{X_u^I}{f^I} - T_x^I) = 0 \qquad \text{(B.14)}$$

$$(r_7^D \frac{Y_u^D}{f^D} - r_4^D)x_w + (r_8^D \frac{Y_u^D}{f^D} - r_5^D)y_w + (r_9^D \frac{Y_u^D}{f^D} - r_6^D)z_w + (T_z^D \frac{Y_u^D}{f^D} - T_y^D) = 0 \qquad \text{(B.15)}$$

$$(r_7^I \frac{Y_u^I}{f^I} - r_4^I)x_w + (r_8^I \frac{Y_u^I}{f^I} - r_5^I)y_w + (r_9^I \frac{Y_u^I}{f^I} - r_6^I)z_w + (T_z^I \frac{Y_u^I}{f^I} - T_y^I) = 0 \qquad \text{(B.16)}$$

en donde los superíndices D e I indican las proyecciones Derecha e Izquierda. En forma matricial

$$[M]\ [a] = [b] \qquad \text{(B.17)}$$

en donde

$$[M] = \begin{bmatrix} r_7^D \frac{X_u^D}{f^D} - r_1^D & r_8^D \frac{X_u^D}{f^D} - r_2^D & r_9^D \frac{X_u^D}{f^D} - r_3^D \\ r_7^I \frac{X_u^I}{f^I} - r_1^I & r_8^I \frac{X_u^I}{f^I} - r_2^I & r_9^I \frac{X_u^I}{f^I} - r_3^I \\ r_7^D \frac{Y_u^D}{f^D} - r_4^D & r_8^D \frac{Y_u^D}{f^D} - r_5^D & r_9^D \frac{Y_u^D}{f^D} - r_6^D \\ r_7^I \frac{Y_u^I}{f^I} - r_4^I & r_8^I \frac{Y_u^I}{f^I} - r_5^I & r_9^I \frac{Y_u^I}{f^I} - r_6^I \end{bmatrix} \qquad [b] = \begin{bmatrix} T_x^D - T_z^D \frac{X_u^D}{f^D} \\ T_x^I - T_z^I \frac{X_u^I}{f^I} \\ T_y^D - T_z^D \frac{Y_u^D}{f^D} \\ T_y^I - T_z^I \frac{Y_u^I}{f^I} \end{bmatrix}$$

$$[a] = \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix}$$

de (B.3) y (B.4)

$$X_u^D = \frac{dpx^D}{sx^D}(Xf^D - Cx^D)\left[1 + k^D\left[\left[\frac{dpx^D}{sx^D}(Xf^D - Cx^D)\right]^2 + \left[dpy^D(Yf^D - Cy^D)\right]^2\right]\right]$$

$$Y_u^D = dpy^D(Xf^D - Cx^D)\left[1 + k^D\left[\left[\frac{dpx^D}{sx^D}(Xf^D - Cx^D)\right]^2 + \left[dpy^D(Yf^D - Cy^D)\right]^2\right]\right]$$

$$X_u^I = \frac{dpx^I}{sx^I}(Xf^I - Cx^I)\left[1 + k^I\left[\left[\frac{dpx^I}{sx^I}(Xf^I - Cx^I)\right]^2 + \left[dpy^I(Yf^I - Cy^I)\right]^2\right]\right]$$

$$Y_u^I = dpy^I(Xf^I - Cx^I)\left[1 + k^I\left[\left[\frac{dpx^I}{sx^I}(Xf^I - Cx^I)\right]^2 + \left[dpy^I(Yf^I - Cy^I)\right]^2\right]\right]$$

La ecuación (B.17) puede resolverse para $[a]$ fácilmente utilizando los métodos convencionales para la solución de sistemas de ecuaciones.

B.6. Apareamiento estéreo

El análisis de movimiento para el apareamiento estéreo es una técnica importante para la correlación de pares de puntos. El movimiento 2D es una proyección del movimiento 3D en escenas reales. Entonces, el modelo de movimiento 2D describe la relación entre regiones de imágenes sucesivas, y proporciona un conjunto reducido de parámetros que permiten encontrar la correspondencia entre pares de imágenes. En nuestro trabajo previo ensayamos con un modelo simplificado de 4 parámetros orientado a regiones, que resulta tener un buen compromiso entre representatividad y economía. Los parámetros del modelo son obtenidos mediante la minimización de estimadores del flujo óptico de máxima probabilidad (estimadores M) robusto y cuadrático, empleando el método del gradiente.

El problema de la estimación de movimiento se puede plantear a través de la figura B.3.

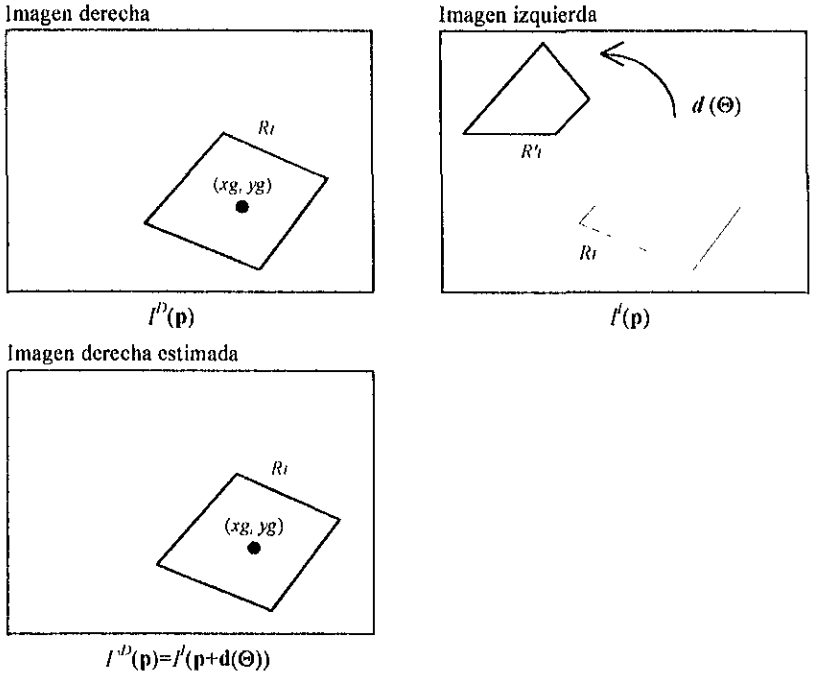


Figura B.3. Análisis de movimiento.

La restricción de conservación de intensidad constante puede plantearse como

$$I'^{D}(\mathbf{p})=I^{I}(\mathbf{p}+\mathbf{d}(\Theta)) \qquad \text{(B.18)}$$

en donde

$\mathbf{p}=(x, y)$ vector de coordenadas

$\mathbf{d}(\Theta)=(dx(\Theta), dy(\Theta))$ vector de desplazamiento

$dx(\Theta)=tx+k(x-xg)-\theta(y-yg)$ desplazamiento en la dirección x

$dy(\Theta)=ty+k(y-yg)+\theta(x-xg)$ desplazamiento en la dirección y

(xg, yg) centro de gravedad de la región Ri

$\Theta=(tx, ty, k, \theta)$ vector de parámetros de movimiento

La ecuación (B.18) se puede resolver para $\mathbf{d}(\Theta)$ mediante la minimización del término de conservación de intensidad constante

$$ED(\mathbf{p}, \mathbf{d}(\Theta))=\rho(I^t(\mathbf{p})-I^{t-1}(\mathbf{p}+\mathbf{d}(\Theta)))$$
$$ED(\mathbf{p}, \mathbf{d}(\Theta))=\rho(DFD(\mathbf{p}, \mathbf{d}(\Theta))) \qquad (B.19)$$

en donde

$DFD(\mathbf{p}, \mathbf{d}(\Theta))= I^t(\mathbf{p})-I^{t-1}(\mathbf{p}+\mathbf{d}(\Theta))$ Diferencia de cuadro desplazado

$\rho(z)$ Estimador

Cuando $\rho(z)=z^2$ esto corresponde al estimador estándar por mínimos cuadrados.

Cuando $\rho(z)=\log\left(1+\frac{1}{2}\left(\frac{z}{\sigma}\right)^2\right)$ esto corresponde al estimador Lorenziano robusto, en donde σ es la desviación estándar.

Resolviendo la minimización de la ecuación (B.19) por el método del gradiente, nos conduce al cálculo iterativo del vector de parámetros

$$\Theta^{t+1}=\Theta-\frac{1}{2Nj}\sum_{\mathbf{p}\in Rj}\varepsilon_\beta \mathbf{G}^t \qquad (B.20)$$

en donde

Nj Tamaño de la región Rj

ε_β Matriz de ganancia adaptable

$$\varepsilon_\beta = \frac{1}{\nabla_x I^{t-1}(\mathbf{p}+\mathbf{d})+\nabla_y I^{t-1}(\mathbf{p}+\mathbf{d})+\alpha}\begin{bmatrix}\varepsilon_c & 0 & 0 & 0\\ 0 & \varepsilon_c & 0 & 0\\ 0 & 0 & \varepsilon_l & 0\\ 0 & 0 & 0 & \varepsilon_l\end{bmatrix}$$

$\varepsilon_c=1$, $\varepsilon_l=0.01$ Parámetros de ponderación

$\alpha=0.001$ Valor para evitar la división entre cero

$\mathbf{G}^t$ Matriz de gradientes

$$\mathbf{G}^t = \begin{bmatrix} \dfrac{\partial}{\partial tx} ED(\mathbf{p},\mathbf{d}) \\ \dfrac{\partial}{\partial ty} ED(\mathbf{p},\mathbf{d}) \\ \dfrac{\partial}{\partial k} ED(\mathbf{p},\mathbf{d}) \\ \dfrac{\partial}{\partial \theta} ED(\mathbf{p},\mathbf{d}) \end{bmatrix}$$

Al resolver las derivadas parciales es fácil demostrar las siguientes soluciones a la ecuación (B.20), para el caso cuadrático y robusto Lorenziano, respectivamente

caso cuadrático

$$\begin{bmatrix} tx^{t+1} \\ ty^{t+1} \\ k^{t+1} \\ \theta^{t+1} \end{bmatrix} = \begin{bmatrix} tx^t \\ ty^t \\ k^t \\ \theta^t \end{bmatrix} - \frac{1}{Nj} \sum_{\mathbf{p}\in Rj} \nabla^2 x I^{t-1}(\mathbf{p}+\mathbf{d}) + \nabla^2 y I^{t-1}(\mathbf{p}+\mathbf{d}) + \alpha \bullet \frac{DFD(\mathbf{p},\mathbf{d})}{} \bullet \begin{bmatrix} \varepsilon_x \nabla x I^{t-1}(\mathbf{p}+\mathbf{d}) \\ \varepsilon_x \nabla y I^{t-1}(\mathbf{p}+\mathbf{d}) \\ \varepsilon_l (x-xg)\nabla x I^{t-1}(\mathbf{p}+\mathbf{d}) + \varepsilon_l (y-yg)\nabla y I^{t-1}(\mathbf{p}+\mathbf{d}) \\ -\varepsilon_l (y-yg)\nabla x I^{t-1}(\mathbf{p}+\mathbf{d}) + \varepsilon_l (x-xg)\nabla y I^{t-1}(\mathbf{p}+\mathbf{d}) \end{bmatrix}$$

caso robusto Lorenziano

$$\begin{bmatrix} tx^{t+1} \\ ty^{t+1} \\ k^{t+1} \\ \theta^{t+1} \end{bmatrix} = \begin{bmatrix} tx^t \\ ty^t \\ k^t \\ \theta^t \end{bmatrix} - \frac{1}{Nj} \sum_{\mathbf{p}\in Rj} [\nabla^2 x I^{t-1}(\mathbf{p}+\mathbf{d}) + \nabla^2 y I^{t-1}(\mathbf{p}+\mathbf{d})] + \alpha \bullet \frac{DFD(\mathbf{p},\mathbf{d})}{} \bullet$$

$$\bullet \frac{1}{[2\sigma^2 + DFD^2(\mathbf{p},\mathbf{d})]} \begin{bmatrix} \varepsilon_x \nabla x I^{t-1}(\mathbf{p}+\mathbf{d}) \\ \varepsilon_x \nabla y I^{t-1}(\mathbf{p}+\mathbf{d}) \\ \varepsilon_l (x-xg)\nabla x I^{t-1}(\mathbf{p}+\mathbf{d}) + \varepsilon_l (y-yg)\nabla y I^{t-1}(\mathbf{p}+\mathbf{d}) \\ -\varepsilon_l (y-yg)\nabla x I^{t-1}(\mathbf{p}+\mathbf{d}) + \varepsilon_l (x-xg)\nabla y I^{t-1}(\mathbf{p}+\mathbf{d}) \end{bmatrix}$$

La desviación estándar σ del estimador robusto generalmente se resuelve en paralelo con la estimación de parámetros. Esta cantidad usualmente es

$\sigma = 1.48\ \text{Med}(|\ Ri - \text{Med}(Rj)\ |)$

en donde Ri y Rj son regiones adyacentes.

B.7. Método

El método involucrado en este proceso de medición de objetos en tres dimensiones es la utilización de la tecnología de visión estéreo para el procesamiento de un par de imágenes o par estéreo, luego por medio de

programación se procesarán las imágenes para la reconstrucción de las medidas reales del objeto bajo inspección.

Las características del software para implementar lo descrito en las secciones 4 (calibración), 5 (reconstrucción) y 6 (apareamiento estéreo) son:

Interface amigable con el usuario (Windows 98).

Programación de métodos numéricos.

Programación de algoritmos para el procesamiento de imágenes en tiempo de adquisición.

B.8. Justificación

La inconveniencia de los métodos tradicionales de medir por coordenadas, sugiere el desarrollo de nuevas técnicas basadas en visión estéreo, para que la medición de objetos resulte una alternativa viable, ya que la tecnología actual en procesamiento de imágenes lo permite. Como parte del proceso, una computadora captura y procesa imágenes rápidamente para así obtener las dimensiones 3D en muy poco tiempo y con una relativa facilidad de medición.

Bibliografía

[1] J .J. Aguilar, F. Torres, M.A. Lope, "Stereo Vision for 3D measurement: accuracy analysis, calibration and industrial applications", *Measurement*, vol. 18 No. 4, pp. 193-200, 1996.

[2] M. J. Black, "Robust Incremental Optical Flow", *PhD thesis*, Yale University, Department of Computer Sciense, USA, September 1992.

[3] F. Bossen, T. Ebrahimi, "A simple and efficient binary shape coding technique based on bitmap representation", *Techical Report MPEG M0964/Tampere*, ISO/IEC JTC1/SC29/WG11, 1996.

[4] S. Burton et al, "MINPACK user's guide" *Argonne National Laboratory*, Minpack Project. march 1980.

[5] K Castleman, *Digital Image Processing*, Prentice Hall, 1996.

[6] D. Cortez, P. Nunes, M. Menezes, F. Pereira, "Image Segmentation Towards New Image Representation Methods", *Signal Processing: Image Communication*, vol. 6, pp. 485-498, 1995.

[7] V. García, "Une approche de compression orientée-objets par suivi de segmentation basée mouvement pour le codage de séquences d'images numériques", *PhD thesis*, Univerité de Rennes Y, Rennes, France, 1995.

[8] R. Gonzalez, R. Woods, *Digital Image Processing*, Addison-Wesley, 1992.

[9] J. D. Helm, M. A. Sutton, S. R. McNeil, "Three-Dimensional Image Correlation for Surface Displacement Measurement", Univerity of South Carolina, Department of Mechanical Engineering.

[10] ISO/IEC JTC1/SC29/WG11 N1727, "MPEG-4 Requirements, version 4 (Stockholm revision)", July 1997.

[11] H. Li, A. Lundmark, R. Forchheimer, "Image Sequence at Very Low Bitrates: A Review", *IEEE Trans. on Image Proc.*, vol. 3, No 5, pp. 589-609, Septiembre 1994.

[12] P. Maragos, R. Schafer, "Morphological Systems for Multidimensional Signal Processing", *Proc of the IEEE*, vol. 78, pp. 690-710, Abril. 1990.

[13] J. H. Moon, J. K. Kim, "On the accuracy and convergence of 2-D motion models using minimum MSE motion estimation", *Signal Processing: Image Communication*, Vol 6, 1994, páginas 319-333.

[14] H. Nicolas, C. Labit, "Motion and illumination variation estimation using a hierarchy of models: application to image sequence coding", *IRISA: Publication Interne No 742*, 1992, 30 páginas.

[15] H. Nicolas, "Hiérarchie de modéles de mouvement et méthodes d'estimation associées: application au codage de séquences d'images", *PhD thesis*, Univerité de Rennes Y, Rennes, France, Septembre 1992.

[16] J. M. Odobez, P. Bouthemy, "Robust Multiresolution Estimation of Parametric Motion Models", *Journal of Visual Communication and Image Representation*, vol. 6, No. 4, pp. 348-365, December 1995.

[17] S. Pateux, C. Labit, "Codage efficace de carte de segmentation pour la compression orientée régions de séquences d'images", *IRISA: Publication Interne No 0000*, 1992, 40 páginas.

[18] W. H. Press et all, *Numerical Recipes in C*, Cambridge University Press, 994 páginas, (2a ed., 1995).

[19] D. F. Rogers, J. A. Adams, *Mathematical Elements for Computer Graphics*, McGraw Hill, 611 páginas, (2a ed., 1990).

[20] P. Salembier, J. Serra, "Flat Zones Filtering, Connected Operators, and Filters by Reconstruction", *IEEE Trans. on Image Proc.*, vol. 4, No 8, pp. 1153-1160, Agosto. 1995.

[21] P. Salembier, M. Pardas, "Hierarchical Morphological Segmentation for Image Secuence Coding", *IEEE Trans. on Image Proc.*, vol. 3, No 5, pp. 639-651, Septiembre 1994.

[22] P. Sander, L. Vincent, L. Cohen, A. Gagalowicz, "Hierarchical Region Based Stereo Matching", *Proceedings of the sixth Scandinavian Conference on Image Analysis*, pp. 71-78, Oulu, Finland, June 1989.

[23] J. Serra. *Image Analysis and Mathematical Morphology*, Academic Press, 1988.

[24] J. P. Tarel, J. M. Vezien, "A Generic Approach for Planar Patches Stereo Reconstruction", *Proceedings of the Scandinavian Conference on Image Analysis*, pp. 1061-1070, Uppsala, Sweden, June 1995.

[25] M. Tekalp, *Digital Video Processing*, Prentice Hall, 526 páginas, (1a ed., 1995).

[26] R. Y. Tsai, "An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Miami Beach, FL, 1986, pages 364-374.

[27] R. Y. Tsai, "A versatile Camera Calibration Technique for High-Accurracy 3D Machine Vision Metrology Using Off-the-Shelf TV

Cameras and Lenses", *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 4, August 1987, pages 323-344.

[28] L. Vincent, "Morphological Grayscale Reconstruction in Image Analysis: Applications and Efficient Algorithms", *IEEE Trans. on Image Proc.*, vol. 2, No 2, pp. 176-201, Abril 1993.

[29] D. Wang, C. Labit, J. Ronsin, "Segmentation-Based Motion Compensated Video Coding Using Morphological Filters", *trabajo a consideración de IEEE Trans. on Circuits & Systems for Video Technology*, presentado en Diciembre 1995.

[30] J. Weng, P. Cohen, M. Herniou, "Camera Calibration with Distortion Models and Accuracy Evaluation", *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 14, No. 10, pp. 965-980, 1992.

[31] L. Wu et al, "Spatio temporal segmentation of image sequences for object-oriented low bit rate image coding", *Signal Processing: Image Communication*, Vol 8, 1996, páginas 513-543.

[32] H. Zheng, S. D. Blostein, "Motion-Based Object Segmentation and Estimation Using the MDL Principle", *IEEE Trans. on Image Proc.*, vol. 4, No 9, pp. 1223-1235, September 1995.