2
Codificação de Textura nos Codificadores de Imagem Orientados por Objeto: as Abordagens SA-DCT e DCT Baseada em Blocos

Até uma década atrás, a compressão de imagens tomava sempre por base o sinal com região de suporte retangular. O padrão MPEG-4 parte 2 extrapolou todos os padrões de compressão de vídeo anteriores ao incorporar o conceito de codificação baseada em objeto [12]-[31]. Esse padrão codifica, decodifica e manipula, de forma independente, segmentos de objetos de vídeo (VOs - *Video Objects*) de forma arbitrária, que são caracterizados por sua forma, textura e movimento. Em conseqüência da ampliação do potencial de aplicação do padrão MPEG-4, a codificação de imagem orientada por objeto tem provocado grande motivação para a pesquisa. Como mencionado no Capítulo 1, o estudo desses codificadores inclui os temas de segmentação e codificação da forma dos objetos, de estimação e compensação de movimento e de codificação de textura. O escopo desta tese restringe-se à codificação de textura de imagens intra-quadros (sem compensação de movimento), em codificadores orientados por objeto.

A Seção 2.1 apresenta um breve resumo dos outros temas relacionados à codificação de imagens baseada em objeto: (i) Segmentação [45]-[65], (ii) Codificação de forma [44],[45],[58],[59] e (iii) Estimação e compensação de movimento para a codificação de textura inter-quadros (preditiva e bi-direcional) [60]-[65]. A Seção 2.2 aborda as principais estratégias em codificação de textura orientada por objeto [66]-[89]. As Seções 2.3 e 2.4 destacam as contribuições mais relevantes apresentadas na literatura em DCT Adaptativa à forma ou SA-DCT (*Shape-Adaptive Discrete Cosine Transform*) [67]-[77] e em DCT baseada em blocos [76]-[83], respectivamente. As abordagens apresentadas nas Seções 2.3 e 2.4 serão tratadas ao longo desta tese.

2.1 Tópicos Relacionados à Codificação de Textura Orientada por Objeto

2.1.1 Segmentação de Imagem

A segmentação de imagem é um tema estudado desde 1985 [45] e consiste na obtenção, no nível compatível com a aplicação visada, de uma partição espacial cujas regiões sejam caracterizadas por conexidade, intensidade correlatada e/ou movimento próprio [44]. As técnicas de codificação de imagens segmentadas não visam apenas à codificação orientada por objeto, mas também à aplicação em algoritmos de compressão que se baseiam na segmentação dos quadros de imagem em regiões homogêneas, com a finalidade única de redução de taxa de bits. Esses algoritmos não suportam, necessariamente, as funcionalidades do MPEG-4 parte 2.

Os algoritmos que têm como propósito a segmentação da imagem em regiões homogêneas (ou planas) visando ao aumento do ganho de codificação utilizam técnicas diversas como (i) a segmentação por limiar de amplitude (para objetos distinguíveis pelas amplitudes de suas intensidades, como imagens binárias, imagens produzidas por sensores infravermelhos e imagens coloridas) [49],[50],[51]; (ii) a segmentação orientada a região, na qual os pixels são inicialmente divididos segundo algum critério (por exemplo, subdivisão em quadrantes ou divisão segundo o gradiente) e posteriormente as regiões vizinhas com características similares são agrupadas segundo uma função-custo [46],[48]; (iii) a segmentação por textura, que se aplica a objetos com padrões de textura altamente regular [53]; (iv) a segmentação por detecção de contornos, que se baseia em algoritmos de detecção de bordas [47],[52] e é adequada à extração de objetos que apresentam alto contraste entre si e em relação ao fundo; (v) a segmentação por casamento de padrões, que utiliza técnicas de reconhecimento de padrões e é especialmente indicada à segmentação de símbolos, caracteres de texto e peças em um ambiente controlado, como uma linha de produção; (vi) a segmentação morfológica, que utiliza filtros 2D ou 3D para simplificação espaço-temporal antes da aplicação dos algoritmos de segmentação propriamente ditos [46],[47] e (vii) a segmentação por análise de movimento, que possibilita separar os elementos móveis dos estáticos a partir da análise de seqüência de quadros em uma cena [47],[53],[61]. A segmentação por análise de movimento em geral proporciona a divisão em um menor número regiões, porque a informação de movimento normalmente é mais homogênea que a informação de textura [61]. Além disso, as regiões segmentadas tendem a

corresponder a objetos reais na cena. Em [54] foram propostas métricas para a análise de eficiência das diversas técnicas de segmentação, levando em conta parâmetros como uniformidade intra-região e contraste inter-regiões, número de objetos na imagem, quantidade e posição dos pixels alocados em regiões não adequadas etc.

A segmentação automática visando à codificação orientada por objeto é uma das tarefas mais difíceis em processamento de imagens, uma vez que pode haver poucas ou nenhuma característica sob controle. A segmentação visando à codificação orientada por objeto, onde um objeto é caracterizado por seu movimento, forma e parâmetros de cor (luminância e crominância), ainda é um tópico aberto à pesquisa. O MPEG-4 parte 2 evita o problema da segmentação usando-se seqüências de vídeo pré-segmentadas (chamadas VOs - *Video Objects*) como entrada do codificador. Foi esse procedimento que permitiu o desenvolvimento dos codificadores orientados por objeto [45].

2.1.2 Codificação de Forma

Entende-se por *forma de um objeto* a descrição geométrica da região de suporte desse objeto, ao passo que o *contorno do objeto* corresponde à curva que, circunscrevendo a sua região de suporte, contém todos os pixels de fronteira do objeto [44]. A codificação de forma é o passo seguinte à segmentação. Nas primeiras aplicações vislumbradas, cada região homogênea demarcada pelo processo de segmentação era codificada transmitindo-se o seu contorno e também o valor da luminância da região [45].

Hoje em dia, a codificação de forma ainda é um tópico especialmente crítico na codificação orientada por objeto. Isso é verdade não somente devido à importância dos contornos na percepção subjetiva da imagem, mas também pela dependência da decodificação de intensidade da imagem em relação à forma do objeto. A principal finalidade da codificação de forma, contudo, não é aumentar a eficiência de codificação, mas possibilitar novas funções e aplicações, como a manipulação independente dos objetos de uma cena [44].

Em [45], relata-se o problema da codificação de forma de objetos em movimento no padrão MPEG-4 parte 2, analisando-se comparativamente vários algoritmos. Inicialmente o padrão selecionou dois modelos para a codificação de forma: o primeiro usa uma aproximação em polígono (codificação baseada em vértice), onde o contorno do objeto a codificar é seguido e a direção de localização do próximo pixel do contorno é codificada.

As diferenças entre os diversos algoritmos baseiam-se no número de pixels vizinhos considerados: 4, 6 ou 8. O segundo modelo é baseado em mapa de bits e atribui o valor '1' ao pixel que pertence ao objeto segmentado. O primeiro modelo mostrou-se mais eficiente na codificação com perda, ao passo que o segundo proporcionou melhores resultados na codificação sem perda. Contudo, o modelo baseado em mapa de bits exige uma carga computacional menor, tendo sido o modelo adotado no padrão MPEG-4 [45].

Em [58], a codificação de forma foi empregada objetivando características de escalabilidade temporal: determina-se individualmente a taxa de transmissão de quadros necessária à recuperação do movimento suave de um objeto específico em uma cena de vídeo. Finalmente, são alocados menos bits para uma região da imagem que apresente movimentos mais lentos, e mais bits para regiões com movimentos mais rápidos.

Mais recentemente, em [59], foi proposto um método para a codificação de forma baseado em vértice que é ótimo no sentido taxa-distorção e leva em consideração a informação de textura do sinal de vídeo de entrada. Esse método utiliza uma banda de tolerância de largura variável, onde a largura é definida segundo as características de textura. Em áreas onde a confiabilidade da estimação da forma e/ou onde erros na definição do contorno não afetem sensivelmente a aplicação, um erro maior de aproximação na codificação de forma é permitido.

2.1.3 Compensação de Movimento

A codificação de vídeo ainda pode explorar as redundâncias temporais através de codificação preditiva interquadros com compensação de movimento, que estima o deslocamento de um quadro para o outro e transmite o vetor de movimento como informação paralela, em adição à imagem de erro de predição [4],[8], [12],[27],[28]. Essa imagem de erro é codificada por um codificador intraquadro que também explora as redundâncias espaciais. É importante ressaltar que para o cálculo da imagem de erro de predição, requer-se a decodificação do quadro a partir do qual será feita a estimação, ainda durante a fase de codificação. Portanto, a carga computacional no codificador e no decodificador não são simétricas.

Para a maioria dos padrões de codificação, três tipos de quadros são definidos: intra (*I-frames*), preditivos (*P-frames*) e bidirecionais (*B-frames*). Um *I-frame* é codificado a cada N quadros, onde N é o tamanho do GOP - *Group Of Pictures*. Um *P-frame* é codificado a partir do *I-frame* ou

do *P-frame* anterior mais próximo e os *B-frames* usam referências tanto passadas como futuras [61]. No MPEG-4 parte 10 ou H.264 AVC, mais de um quadro anterior podem ser usados para a estimação de um *P-frame* e nesse caso, são transmitidos os parâmetros de referência desses quadros. Esse padrão ainda permite a codificação de um *P-frame* no modo *SKIP*. Nesse modo, não são transmitidos nem o sinal de erro de predição quantizado, nem o vetor de movimento, nem o parâmetro de referência. O sinal reconstruído é obtido a partir do sinal de referência localizado na posição 0 do *buffer* de quadros e o vetor de movimento pode ser nulo.

Em [61] é apresentado um algoritmo para a compensação de movimento adaptativa à região, oferecendo uma solução para as exigências de interatividade com o conteúdo visual do padrão MPEG-4 parte 2. São testados três possíveis vetores de predição para estimar o movimento de cada pixel, um em cada escala de resolução, sendo selecionado aquele que minimizar o erro de predição. A característica de escalabilidade temporal desse algoritmo permitiu trabalhar com taxas de transmissão de quadros mais altas (60 Hz). Contudo, resultados experimentais sobre imagens da classe C (médio nível de detalhes e grande intensidade de movimento ou *vice-versa*) mostraram que o algoritmo de compensação de movimento adaptativo à região proporcionou resultados apenas comparáveis ao algoritmo implementado no padrão MPEG-1 [61].

Em [60] foi proposta a implementação de um '*toolbox*' de algoritmos de compensação de movimento orientada por objeto. A idéia é que após o processo de segmentação cada objeto seja classificado sob alguns aspectos de movimento, após uma análise preliminar da cena. Por exemplo, cena fixa com a câmera em movimento, objetos rígidos em movimento, objetos flexíveis em movimento etc. Essa classificação é utilizada para selecionar a ferramenta de predição de movimento mais adequada. Os autores sugerem a ampliação desse pacote de ferramentas com a introdução de futuros algoritmos de compensação de movimento.

Apesar de todo esse esforço em pesquisa de algoritmos de compensação de movimento adaptativa ao objeto segmentado, o algoritmo de compensação de movimento selecionado para o MPEG-4 é baseado em blocos [45]. Essa opção deve-se a dois motivos: primeiro, porque a complexidade computacional é menor e segundo, porque não se nota diferença na qualidade subjetiva da imagem, como já havia sido comentado em [61].

O trabalho [63] aborda a questão do atraso introduzido devido à codificação dos *B-frames*, tendo sido mostrado que para imagens com grande intensidade de movimento, os benefícios relacionados à redução da taxa

de bits obtida com a predição bidirecional não compensam o aumento do atraso. O esquema de predição bidirecional introduz um atraso significativo porque o *B-frame* necessita ser lido antes de ser codificado. Por exemplo, usando um padrão de codificação IBBBPBBBPBBB (3 *B-frames* entre um *I-frame* ou um *P-frame*), o atraso introduzido corresponde a quatro quadros. No decodificador, o atraso é de apenas um quadro, uma vez que os *B-frames* são transmitidos após os quadros *I/P* relevantes.

Em [62] foi feita uma análise comparativa entre seis algoritmos de compensação de movimento baseados em bloco. A comparação foi realizada levando-se em conta o desempenho e a carga computacional exigida pelos algoritmos.

Em [64] foi apresentado um novo método de estimação de movimento baseada em objeto para computar os *B-frames*, assumindo que o vetor de movimento seja decomposto em dois. O primeiro refere-se ao movimento rígido e é a parcela dominante, ao passo que o segundo refere-se à deformação residual e tem por objetivo o refinamento do movimento de objetos flexíveis. O algoritmo mostrou-se eficiente na estimação de quadros onde há regiões com diferentes intensidades de movimento. Já em [65], foi apresentado um algoritmo de codificação de vídeo a taxas de bits múltiplas. Nesse algoritmo, a estimação de movimento é realizada no domínio da transformada. Assim, elimina-se a necessidade da DCT inversa na codificação. Foi mostrado que o algoritmo proporciona uma redução significativa da carga computacional exigida no codificador, sob a pena de redução da RPR (Razão Pico/Ruído) em apenas 0,3 dB.

2.2 As Principais Abordagens em Codificação de Textura Orientada por Objeto

A codificação orientada por objeto é uma das características mais importantes introduzidas pelo MPEG-4 parte 2. Ao codificar um objeto de forma arbitrária ao invés de usar uma região de suporte retangular, o MPEG-4 permite manipular e interagir com os objetos depois que eles foram criados e codificados.

Com o objetivo de prover as funcionalidades baseadas em objeto do padrão MPEG-4 parte 2, novas técnicas de codificação de textura têm sido desenvolvidas para a descrição de regiões de imagem de forma arbitrária. No padrão MPEG-4 parte 2, tanto a Wavelet como os esquemas utilizando DCT estão previstos para a codificação de textura de um objeto parado,

mas somente a DCT é empregada para a codificação de objetos em movimento [87]. Nos esquemas utilizando a DCT, o objeto de forma arbitrária primeiro é inscrito em uma região retangular, que então é particionada em blocos. Os blocos inteiramente contidos no objeto são codificados através do método tradicional (DCT-2D) e os blocos inteiramente fora do objeto são descartados. Os blocos parcialmente preenchidos por pixels do objeto são codificados da seguinte forma [87]: os *I-frames* podem usar o algoritmo de extrapolação LPE - *Low Pass Extrapolation* seguido de DCT [17],[18],[82] ou a $\triangle DC - SA - DCT$ [70]. Já os *P-frames* e os *B-frames* podem usar o algoritmo de extrapolação '*zero-padding*' [82] seguido de DCT ou a SA-DCT [67].

Um esquema baseado em Wavelet também foi proposto para codificar objetos parados de forma arbitrária no MPEG-4 parte 2. Nesse caso, o objeto não é particionado em blocos; ao invés disso, o objeto inteiro primeiro é submetido a uma transformada Wavelet adaptativa à forma (SA-DWT) com coeficientes alinhados pela fase [85]. Os coeficientes Wavelet em seguida são quantizados usando o algoritmo EZW [91] e codificados por entropia. O algoritmo *EZW - Embedded Zerotree Wavelet*, proposto em 1993, pode terminar a codificação a qualquer ponto, permitindo a medida exata da distorção. Além disso, para uma taxa de bits requerida, o decodificador pode parar a decodificação a qualquer ponto da seqüência, produzindo uma imagem que seria exatamente igual àquela codificada com a taxa de bits correspondente da seqüência de bits truncada.

Na implementação da Transformada Wavelet Discreta (DWT) usando bancos de filtros bidimensionais separáveis, dois filtros 1-D (um passa-baixas e outro passa-altas) são usados para gerar quatro filtros bidimensionais separáveis: um passa-baixas em ambas as freqüências espaciais (a banda LL), um passa-altas em ambas as freqüências espaciais (a banda HH) e dois filtros que são passa-baixas em uma freqüência espacial e passa-altas na outra (bandas LH e HL). Após essa decomposição espectral da imagem inicial e posterior decimação, a subbdanda de baixas freqüências é sucessivamente decomposta em componentes de menor resolução, gerando uma decomposição final em $7, 10, 13 \cdots$ subbandas, como mostra a Figura 2.1, onde o caso de 7 subbandas foi ilustrado [88].

No caso de imagens retangulares, a decomposição em subbandas pode ser usada para explorar a compactação de energia na subbanda de baixas freqüências e naquelas subbandas que concentram a informação espectral correspondente às direções de borda dominantes. Para a SA-DWT, o número de bordas significativas é muito menor do que no caso de imagens regulares, uma vez que somente um objeto é codificado e, portanto, não existem

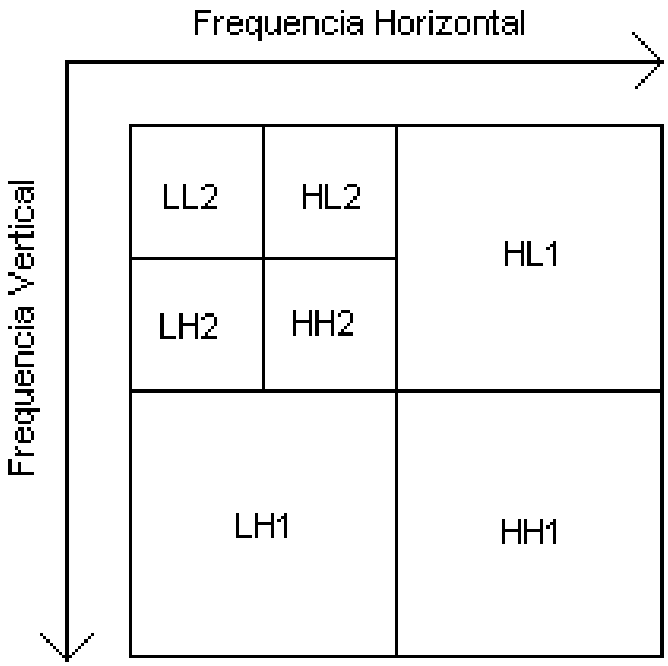

Figura 2.1: Decomposição em 7 subbandas

bordas. Além disso, a intensidade do objeto tende a apresentar variações mais suaves. Dessa forma, é justo assumir que, para a codificação orientada por objeto, a energia estará ainda mais concentrada nas subbandas de baixa freqüência [88].

Os esquemas apresentados na literatura sobre SA-DWT apresentam alternativas na forma de calcular os pixels externos a serem inseridos no objeto original durante o processo de decimação [84]-[88].

Em [89], os autores apresentaram um algoritmo capaz de reconstruir apenas uma região de interesse antes que o resto da imagem seja recuperado. Essa funcionalidade foi denominada *codificação ROI - Region Of Interest*. O algoritmo aplica o algoritmo de particionamento em árvores hierárquicas SPIHT - *Set Partitioning In Hierarchical Trees* [92] para codificação de imagens usando Wavelet, modificando o ordenamento do algoritmo SPIHT de forma a atribuir uma ênfase maior aos coeficientes da transformada que pertençam à ROI. Essa funcionalidade pode ser bastante útil em aplicações como *web browsing*, acesso a databases e telemedicina.

Vale lembrar que, conforme foi mencionado no Capítulo 1, o algoritmo MPEG-2 escalável multi-resolução (diferentemente do MPEG-2 escalável SSP), utiliza a decomposição em subbandas para proporcionar a escalabilidade espacial [29]. A camada-base é constituída pela subbanda LL e a

camada de detalhes, pelas subbandas LH, HL e HH.

A SA-DCT [67] é uma outra abordagem utilizada em codificação de imagens orientada por objeto. Ela é baseada em um conjunto pré-definido de funções-base DCT separáveis e representa um bom compromisso entre complexidade de implementação, eficiência de codificação e compatibilidade com as técnicas DCT existentes. É capaz de codificar objetos de forma arbitrária dentro de blocos de imagem de dimensão $N \times N$, proporcionando bons ganhos de eficiência de codificação, especialmente a taxas altas.

No algoritmo SA-DCT, os blocos de contorno de dimensão $N \times N$ são transformados em segmentos verticais de comprimento variável L $(1 \leq L \leq N)$, que contêm apenas pixels pertencentes ao objeto. Em seguida, uma transformada unidimensional de tamanho L pré-definida é aplicada a cada segmento. Num segundo estágio, os coeficientes verticais resultantes são alinhados pelo índice, resultando em segmentos horizontais de tamanho variável L. Esses segmentos finalmente são processados na direção horizontal, aplicando-se transformadas unidimensionais de tamanho L sobre cada um deles [67].

Embora a SA-DCT seja uma ferramenta importante para a compressão de vídeo baseada em objeto, ela é definida por um conjunto de funções-base de tamanhos diferentes, tornando difícil o uso de algoritmos rápidos e o projeto de circuitos integrados dedicados (algoritmos rápidos para uma DCT de K pontos normalmente não existem, se K não é uma potência de 2). Sendo assim, uma opção atraente e alternativa à SA-DCT é o uso da DCT baseada em blocos associada aos algoritmos de extrapolação [78]-[81], permitindo o uso direto dos algoritmos rápidos de DCT-2D (para a implementação do *software*) e dos circuitos integrados comercialmente disponíveis (para a implementação do *hardware*). Nessa abordagem, os blocos de contorno são pré-processados, inicializando-se os pixels externos ao objeto antes da aplicação da DCT-2D. A idéia básica consiste em empregar expansões suaves dos pixels do objeto, tornando uniforme e fortemente correlatada a distribuição de amplitudes nos blocos de contorno, de forma que a energia dos coeficientes da DCT possa ser concentrada nas componentes de baixa freqüência. Embora o número de coeficientes DCT a serem calculados e quantizados após o processo de extrapolação seja maior que no caso da SA-DCT, espera-se um ganho de velocidade no cálculo da transformada, já que podem ser utilizados algoritmos rápidos de implementação.

2.3 A SA-DCT

Em 1995, Sikora e Makai apresentaram um algoritmo DCT capaz de codificar objetos de forma arbitrária dentro de blocos de imagem $N \times N$, chamado SA-DCT (*Shape Adaptive DCT*) [67], como já comentado na seção anterior. Esse algoritmo pode ser visto como uma aproximação do método descrito por Gilge em [66], com uma complexidade computacional muito menor. A imagem é separada em blocos $N \times N$ adjacentes e somente os blocos inteiramente contidos em um objeto ou região segmentada são codificados usando a DCT-2D $N \times N$. Os blocos que contêm contornos das regiões segmentadas são codificados usando o algoritmo proposto em [67]. Esse algoritmo transforma os segmentos de imagem em segmentos de direção horizontal e vertical separadamente, aplicando um conjunto de funções-base DCT unidimensionais pré-definidas, iguais a $S_L.[DCT-L]$, onde a matriz $[DCT-L]$ é dada por

$$[DCT-L] = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \cdots & \frac{1}{\sqrt{2}} \\ \cos\frac{\pi}{2L} & \cos\frac{3\pi}{2L} & \cdots & \cos\frac{(2L-1)\pi}{2L} \\ \vdots & \vdots & & \vdots \\ \cos\frac{(L-1)\pi}{2L} & \cos\frac{3(L-1)}{2L} & \cdots & \cos\frac{(2L-1)(L-1)\pi}{2L} \end{bmatrix} \quad (2\text{-}1)$$

Em um segundo estágio, os coeficientes da SA-DCT verticais de mesmo índice (por exemplo, todos os coeficientes DC, depois todos os primeiros coeficientes AC etc) são transformados na direção horizontal, novamente usando a matriz $S_L.[DCT-L]$. No caso do algoritmo de Sikora, a transformação $S_L.[DCT-L]$ é ortogonal não normalizada (NO-SA-DCT). Faz-se $S_L = 2/L$, de forma que o produto $S_L L$ seja constante. Isso garante que o valor médio da imagem seja completamente mapeado no coeficiente DC.

A Figura 2.2 ilustra as transformações da região de suporte na aplicação da SA-DCT: (a) o bloco de contorno original com 6 segmentos verticais de tamanhos 1, 2, 3, 4, 6 e 3, respectivamente; (b) alinhamento dos segmentos verticais resultantes do primeiro processamento unidimensional na direção vertical (alinhamento pelo índice, correspondendo a deslocar os segmentos verticais em direção à borda superior do bloco); (c) coeficientes horizontais resultantes do segundo processamento unidimensional, na direção horizontal; (d) alinhamento dos coeficientes finais na borda esquerda do bloco.

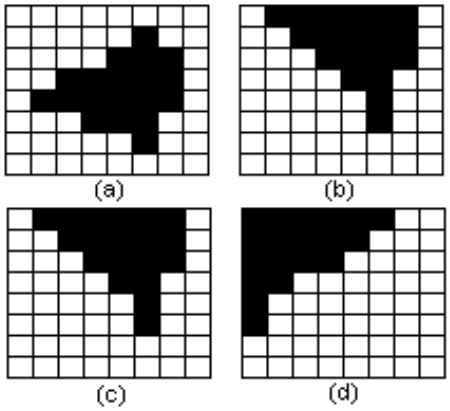

Figura 2.2: Transformações da região de suporte na aplicação da SA-DCT

Os coeficientes finais da SA-DCT podem ser quantizados e codificados por um código de comprimento de corrida em um esquema de codificação híbrido padrão, como o MPEG-1. Os parâmetros de contorno são transmitidos ao receptor como informação paralela e codificados sem perda.

Ainda em 1995, Sikora et al apresentaram uma análise de transformadas bidimensionais adaptativas à forma [68]. Foi ressaltado que a eficiência de uma transformada aplicada aos pixels em um bloco retangular depende das propriedades de correlação entre os pixels. Apresentou-se a SA-KLT (*Shape-Adaptive KLT*) bidimensional, muito superior em desempenho à Transformada de Gilge, mas requerendo uma carga computacional equivalente. Também foi mostrado experimentalmente que a SA-DCT de Sikora e Makai apresenta desempenho de codificação comparável ao desempenho da Transformada de Gilge, sob um custo computacional muito inferior.

Em 1996, Bi et al [69] publicaram uma severa crítica ao trabalho de Sikora no que diz respeito à forma de agrupamento dos coeficientes verticais para o segundo estágio da transformação (alinhamento pelo índice). Foi mostrado experimentalmente que a correlação cruzada dos coeficientes da SA-DCT semi-transformados na direção vertical é fortemente dependente do comprimento das colunas e da distância entre seus centros. Segundo os autores, o alinhamento dos coeficientes verticais para a aplicação do segundo estágio da transformada requereria uma análise mais apurada, que dependeria da correlação real entre os coeficientes da SA-DCT verticais, o que aumentaria muito a complexidade da implementação. Uma possível solução seria a utilização de uma tabela-consulta, que armazenaria os pares de coeficientes de máxima correlação.

Outra restrição à SA-DCT foi apresentada em 1997 por Kauff e Schurr [70], referindo-se à opção desses autores em utilizar DCTs uni-dimensionais não normalizadas para evitar a distorção do valor médio da intensidade da imagem. Se o valor médio no domínio espacial não é mapeado diretamente no coeficiente DC no domínio da transformada, um nível de cinza constante pode ser transformado em coeficientes AC adicionais e então, após a quantização dos coeficientes e o cálculo da SA-DCT inversa, o padrão de cinza reconstruído é degradado. Esse fenômeno é denominado 'distorção da média' da SA-DCT e não acontece na transformada NO-SA-DCT (SA-DCT não normalizada) de Sikora. Em contrapartida, essa transformada pode ser apenas sub-ótima sob o ponto de vista de desempenho de codificação, especialmente se for aplicada a regiões de tamanhos diferentes. Nesse caso, o erro médio quadrático - *emq* - do ruído de quantização não é totalmente controlado pelo quantizador, dependendo também do tamanho da região da imagem. Esse efeito é conhecido por 'distorção da correlação cruzada do ruído de quantização'.

Teoricamente, só existe uma situação - a de média zero - onde ambos os efeitos são evitados simultaneamente. Nesse caso, a 'distorção da média' não ocorre e a PO-SA-DCT (SA-DCT pseudo-ortonormal) pode ser aplicada à imagem de média nula para evitar a *'distorção da correlação cruzada do ruído de quantização'*. Sob a pré-condição de média nula, a eficiência da PO-SA-DCT supera a eficiência da NO-SA-DCT. Como solução, Kauff e Schurr propõem a ΔDC-SA-DCT, onde o codificador calcula individualmente o valor médio m de cada região e a seguir o valor m é subtraído de todos os pixels daquela região, sendo a PO-SA-DCT aplicada à imagem resultante. Contudo, o problema resultante é que a saída da PO-SA-DCT não é compatível com a DCT padrão, porque o coeficiente (1,1) não representa o valor DC da imagem original. Para solucionar esse problema, o valor DC é substituindo pelo coeficiente (1,1) que seria resultante da DCT padrão. Depois dessa manipulação, a imagem transformada apresenta coeficientes DC e AC adequados, que podem ser codificados usando quantização convencional e técnicas de codificação de comprimento variável.

Ainda em 1997, Moon et al propuseram um esquema de particionamento da imagem adaptativo à forma do objeto [76], a fim de reduzir o número total de blocos a serem codificados, com vistas à utilização conjunta com a SA-DCT. Assim, ao invés de sempre se adotar como referência para a primeira camada (linha) de blocos o pixel localizado no canto superior esquerdo do quadro, essa referência é calculada para cada objeto a ser

codificado.

Em 2000, NG e Lin propuseram uma nova variação da SA-DCT [72]. O algoritmo é uma combinação da técnica BBM [71] com um algoritmo modificado da SA-DCT, sendo chamado de BBGM-SA-DCT *(Boundary-Block Group and Merging)*. A técnica *BBM* foi apresentada em 1998 para a codificação orientada por objeto usando DCT baseada em blocos e consiste em unir pares de sub-blocos vizinhos pré-definidos (na horizontal, vertical ou diagonal), localizados em um mesmo macrobloco, reduzindo-se o número de blocos a serem processados [71]. A técnica BBGM, por sua vez, consiste em agrupar blocos de contorno vizinhos obedecendo a certas propriedades, produzindo a partir deles um único bloco. O algoritmo de SA-DCT modificado realiza ajustes dos coeficientes DC, adaptativamente ao número de pixels. Além disso, a técnica BBGM proporciona o agrupamento de 2 a 4 sub-blocos 8×8 vizinhos, enquanto que o número de blocos a serem agrupados através do método BBM [71] é restrito a dois. A grande vantagem, contudo, é que a forma do segmento de bloco não restringe o agrupamento como na técnica BBM.

Ainda em 2000, Shen, Zeng e Liou [77] apresentaram um método simples e eficiente para aumentar a eficiência de codificação da SA-DCT, a partir da escolha da direção preferencial de processamento do algoritmo (horizontal ou vertical).

Também em 2000, em [44], Acocella apresentou analiticamente o problema da distorção da média por efeito do erro de quantização, sob três condições distintas. A primeira refere-se a imagens com valor médio nulo, para as quais não ocorre distorção da média por efeito do erro de quantização (devido à parcela dos coeficientes da transformada que estão associados ao valor médio da imagem), correspondendo ao tratamento adotado por Kauff e Schurr em [70]. A solução permite a adoção da transformada pseudo-ortonormal (*PO-SA-DCT*), com $S_L = \sqrt{\frac{2}{L}}$, evitando o problema da correlação cruzada do ruído de quantização. Contudo, mostrou-se que o erro decorrente da quantização dos coeficientes AC da imagem também provoca distorções de média, o que não havia sido considerado em [70].

A segunda condição refere-se às transformadas onde $S_L L$ é constante (como no algoritmo SA-DCT de Sikora em [67], com $S_L = \frac{2}{L}$), onde o valor médio da imagem é completamente mapeado no coeficiente DC. A sua quantização produz uma variação uniforme do valor médio em toda a imagem após a decodificação, não ocorrendo, portanto, degradação significativa. A distorção decorrente da parcela AC da imagem continua ocorrendo, mas o inconveniente mais grave é o problema da correlação do

ruído de quantização, conforme já havia sido mostrado em [70].

A última condição refere-se a imagens com valor médio distinto de zero utilizando a transformada com $S_L L \neq constante$. A distorção do valor médio decorre tanto do valor médio como da parcela AC da imagem, cujos erros de quantização provocam conjuntamente uma variação não uniforme do valor médio da imagem.

Acocella em [44] propõe um método para eliminar ou minimizar os efeitos não só da distorção do valor médio, como também da parcela AC da imagem, garantindo que a imagem obtida na decodificação, após a quantização, permaneça com média nula.

Na etapa de codificação do método proposto em [44], calcula-se o valor médio, que então é subtraído da intensidade da imagem, como proposto por Kauff e Schurr em [70]. Em seguida, realiza-se a quantização uniforme dessa média com 8 bits. São obtidos os coeficientes da *PO-SA-DCT* da nova imagem de média nula (com $S_L = \sqrt{\frac{2}{L}}$) e os coeficientes, com distribuição de probabilidade laplaciana, são quantizados por um quantizador escalar não uniforme com atribuição dinâmica de bits, em função da variância de cada coeficiente. Finalmente, faz-se a codificação de Huffman dos coeficientes da imagem de média nula e reúne-se a média quantizada e os coeficientes codificados da imagem de média nula em uma seqüência de bits única.

Na fase de decodificação do método proposto em [44], separa-se a média quantizada da seqüência de bits, decodifica-se por Huffman os coeficientes da imagem de média nula, obtêm-se as linhas de coeficientes após a aplicação da IDCT-1D horizontal sobre os coeficientes quantizados e substitui-se a primeira linha dos coeficientes resultantes por um vetor-linha de distância mínima em relação ao anterior, que não viole os níveis de quantização utilizados e atenda à condição definida por

$$[\sqrt{L_1}\ \sqrt{L_2}\ \ldots\ \sqrt{L_m}] \cdot \underline{b}_1'^t = 0 \qquad (2\text{-}2)$$

Essa condição garante que, ao se utilizar $S_{L_i} = \sqrt{\frac{2}{L_i}}$, a imagem decodificada obtida dos coeficientes quantizados apresente valor médio nulo. Em seguida, aplica-se a IDCT vertical, obtendo-se a imagem de média nula decodificada, e por último adiciona-se a média quantizada à imagem de média nula decodificada.

Em [44], Acocella abordou também o problema da correlação cruzada do ruído de quantização. Foi mostrado que o fator $S_L = \frac{2}{L}$ empregado por Sikora em [67] elimina a distorção do valor médio, mas colore o erro de intensidade da imagem, mesmo quando o ruído de quantização é

branco, tornando o erro da intensidade da imagem dependente da forma da imagem. Foi mostrado um estudo analítico da correlação cruzada do ruído de quantização ao se empregar os fatores S_L mais utilizados na literatura, tendo sido mostrado que o método proposto em [44] também elimina o inconveniente desse problema.

Shen et al publicaram em 2000 um trabalho onde propuseram um esquema híbrido para uma implementação alternativa da SA-DCT [73], buscando a diminuição da complexidade de implementação do algoritmo original, mantendo a eficiência de codificação. O método consiste em selecionar uma dentre duas possíveis estratégias para processar um segmento de tamanho L em um bloco de contorno. A primeira estratégia é um método de extrapolação que apresenta a propriedade de preservação da forma, que se caracteriza por produzir um segmento de tamanho N no domínio espacial que tenha $N - L$ coeficientes nulos no domínio da transformada. A segunda estratégia é a aplicação da matriz de transformação $S_L.[DCT - L]$, como faz o algoritmo de Sikora, sobre o segmento de tamanho L. A escolha entre uma e outra estratégia depende do tamanho L do segmento: para $1 \leq L \leq 5$, a complexidade da SA-DCT é menor, mas para $6 \leq L < N$, a estratégia alternativa envolve uma complexidade computacional menor. Isso ocorre porque para processar um segmento de tamanho L usando a matriz $S_L.[DCT - L]$, são necessárias L^2 multiplicações e $L(L-1)$ adições. Já o processamento desse segmento com o algoritmo de extrapolação alternativo consiste de duas partes: inicialmente, a fase de extrapolação, englobando $L(N-L)$ multiplicações e $L(N-1-L)$ adições. Posteriormente, a aplicação da DCT-2D de dimensão N, para a qual existem algoritmos rápidos no caso de N ser uma potência de 2.

Em 2001, Acocella e Alcaim [74] abordaram o problema do alinhamento ótimo dos coeficientes DCT de colunas distintas para o cálculo dos coeficientes DCT horizontais na implementação da SA-DCT, problema esse abordado experimentalmente em [69]. Os autores em [74] propuseram um método analítico de alinhamento dos coeficientes verticais pela fase, em contrapartida ao alinhamento pelo índice, visando à melhoria de desempenho no que diz respeito à concentração de energia e descorrelação dos coeficientes finais da transformada. Da análise dos resultados obtidos, constatou-se que para taxas baixas, o método original de Sikora, que utiliza o alinhamento dos coeficientes verticais pela ordem [67], continua sendo o mais adequado. Para aplicações que empregam taxas mais elevadas, o método proposto mostrou-se mais eficiente, atribuindo-se a causa do mau funcionamento a taxas baixas à alteração da correlação cruzada pela quantização

escalar dos coeficientes da SA-DCT.

Acocella e Alcaim desenvolveram também em 2001 uma formulação matemática [75] para transformadas bi-dimensionais adaptativas à forma, onde a SA-DCT é um caso particular. Os autores mostraram que através de um simples mapeamento de $\Re^2$ em $\Re$, uma transformada adaptativa à forma pode ser escrita em termos da concatenação de três operadores lineares ($T = T_3T_2T_1$). O operador T_1 gera os coeficientes verticais das transformadas, o operador T_2 realiza a permutação dos coeficientes verticais de acordo com o procedimento de alinhamento adotado (pela ordem [67], pela fase [74] ou outro) e o operador T_3 produz os coeficientes horizontais das transformadas. A formulação apresentada neste trabalho fornece uma base importante para futuros desenvolvimentos em codificação de vídeo orientada por objeto.

2.4 A DCT Baseada em Blocos

Como foi comentado na Seção 2.2, uma opção atraente e alternativa à SA-DCT é o uso da DCT baseada em blocos associada aos algoritmos de extrapolação [78]-[81], permitindo o uso direto dos algoritmos rápidos de DCT-2D. Nessa abordagem, os blocos de contorno são pré-processados, inicializando-se os pixels externos ao objeto antes da aplicação da DCT-2D.

Seguindo essa abordagem, em 1995 Cho et al apresentaram em [78] um algoritmo de extrapolação para segmentos de imagem de forma arbitrária. Esse algoritmo foi denominado *Extension/Interpolation* (EI) e é constituído de dois procedimentos uni-dimensionais, o primeiro aplicado em uma direção e o segundo na outra. Associado à DCT baseada em blocos, esse esquema de codificação será denominado, neste texto, EI-DCT.

Para ilustrar a EI-DCT, consideremos que primeiro as operações sejam feitas na direção horizontal e depois, na direção vertical. Para cada uma das S linhas de objeto de tamanho M (M varia de acordo com a linha) em um bloco retangular de comprimento N, onde ($1 \leq M < N$), calcula-se a DCT-1D sobre seus M pixels, obtendo-se M coeficientes da transformada. Em cada linha não nula, são inseridos ($N - M$) zeros no domínio da transformada e aplicada a DCT inversa sobre os N pontos, resultando N pixels interpolados no domínio espacial. Ao final do processamento na direção horizontal, resultarão S linhas de tamanho N. A seguir, o mesmo procedimento unidimensional é aplicado sobre as N colunas de tamanho

S. Esse método não introduz componentes de alta freqüência, mas requer elevado esforço computacional.

A Figura 2.3 ilustra um exemplo do processo de extrapolação, na etapa de codificação, e do processo inverso de extrapolação, obrigatório do algoritmo EI, na etapa de decodificação.

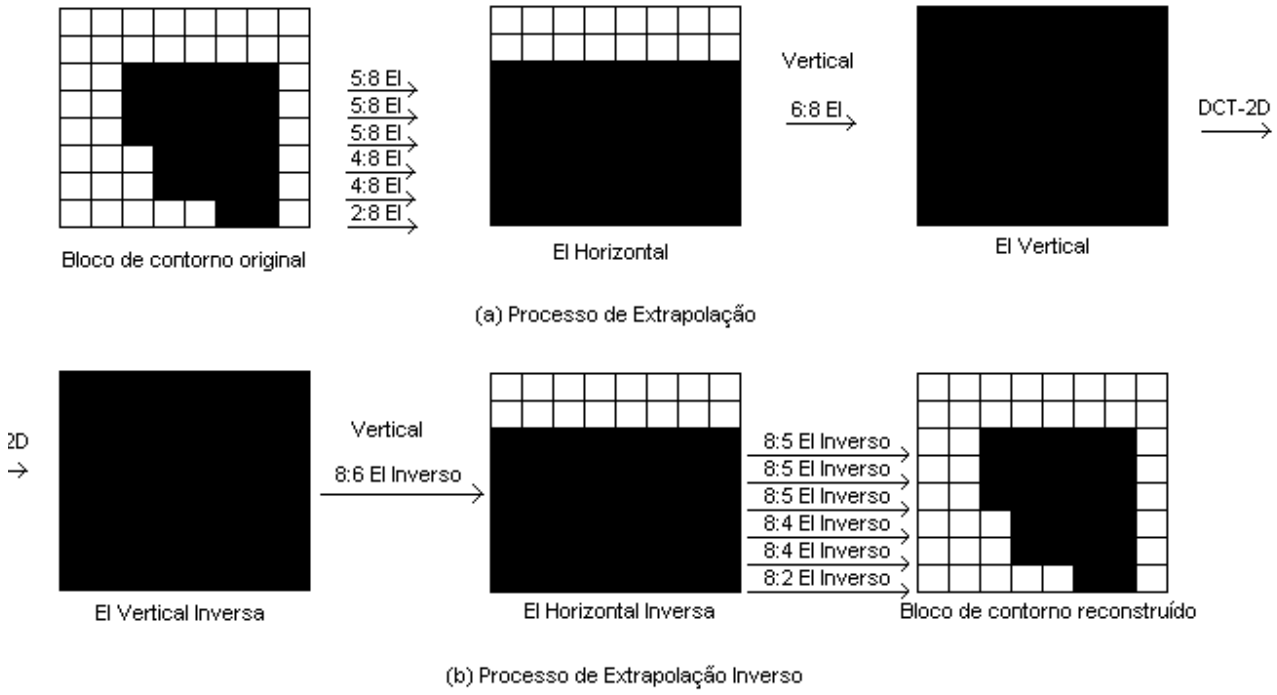

Figura 2.3: Bloco de contorno extrapolado usando o algoritmo EI. (a) Processo de extrapolação (codificador) e (b) Processo de extrapolação inverso (decodificador)

Em 1997, Xie et al apresentaram um método de extrapolação para a DCT baseada em blocos, denominado *Expanded Arbitrary-Shaped DCT* (EA-DCT) [79]. O método pode necessitar de vários passos de cálculo para a determinação dos valores de todos os pixels a serem extrapolados. Em cada passo, apenas os pixels fora do objeto segmentado, mas que sejam vizinhos a ele de acordo com o critério dos 8 vizinhos (V_1 a V_8), como mostrado na Figura 2.4, são extrapolados. Para realizar a extrapolação, é atribuído ao pixel P o valor da média aritmética dos pixels vizinhos a ele que pertençam à segmentação. Após a sua extrapolação, o pixel é incorporado à região segmentada, antes do início do próximo passo.

Para reduzir a complexidade computacional, um processo de extrapolação simplificado foi apresentado ainda em [79]. A idéia principal é extrapolar todos os pixels externos ao objeto em um único passo, substituindo-os pelo valor médio dos pixels do objeto, sob a hipótese de que a aproximação

V_1	V_2	V_3
V_4	P	V_5
V_6	V_7	V_8

Figura 2.4: Os oito vizinhos do pixel P a ser extrapolado

de intensidade uniforme é aceitável em blocos pequenos, como os de tamanho 8×8. Esse método foi denominado *Simplified EA-DCT* (SEA-DCT).

Em 1998, Yi et al [80] apontaram o aumento significativo dos coeficientes de alta freqüência como uma desvantagem dos métodos de extrapolação EA-DCT e SEA-DCT apresentados em [79], ao serem comparados ao método EI-DCT [78], que pela natureza do processamento não introduz componentes de alta freqüência e é mais eficiente. Em contrapartida, apresenta como desvantagem a grande complexidade computacional. Para reduzir o esforço de cálculo do método EI-DCT, os autores propõem em [80] uma implementação rápida alternativa, baseada puramente em operações matriciais. A idéia é calcular previamente e armazenar os coeficientes resultantes da transformação $A^{-1}B$, tal que $F_2 = A^{-1}BF_1$, onde:

- F_1: segmento dos M pixels originais, contidos no interior do objeto.
- F_2: segmento dos N pixels interpolados.
- B: DCT unidimensional aplicada sobre um segmento de tamanho M.
- A^{-1}: DCT inversa unidimensional aplicada sobre um segmento de tamanho N.

Ao invés de processar inicialmente uma direção arbitrária como proposto no método original [78], os autores propõem ainda [80] que a direção preferencial para o primeiro procedimento unidimensional na EI-DCT seja determinada pelas variâncias dos comprimentos das linhas e das colunas do objeto segmentado em cada bloco, devendo o procedimento ser aplicado primeiro na direção que apresentar a menor variância. O esquema de particionamento ótimo de blocos apresentados em [76] também foi utilizado em associação ao método EI, objetivando reduzir o número de blocos a serem codificados e, conseqüentemente, diminuir a taxa de bits.

A técnica de particionamento *SARP* proposta em [76] foi implementada no modelo de verificação 7.0 do MPEG-4 e utilizada para determinar o retângulo ótimo que irá circunscrever o objeto segmentado [25]. Esse

retângulo deve conter o menor número possível de macroblocos 16×16, de forma a melhorar a eficiência de codificação. A referência superior esquerda do retângulo move-se para a esquerda e/ou para cima, de 0 a 15 pixels, em passos de 2 pixels. Poderão ocorrer três tipos possíveis de macroblocos: os internos, contendo apenas pixels do objeto; os externos, que não contêm nenhum pixel do objeto, e os de contorno, contendo tanto pixels do objeto quanto pixels do fundo. A textura de cada macrobloco é codificada de acordo com o tipo do macrobloco. Os macroblocos externos à região segmentada não são codificados. Os macroblocos internos à região segmentada são divididos em 4 sub-blocos de tamanho 8×8 e cada sub-bloco é codificado como nos padrões MPEG-1 e MPEG-2. Nos macroblocos de contorno, pode haver três tipos de sub-blocos: externos à região segmentada (não codificados), internos à região segmentada (codificados como no MPEG-1 e MPEG-2) e de contorno. Para os sub-blocos de contorno, a região do fundo é preenchida com valores apropriados (resultantes da extrapolação), antes da aplicação da DCT baseada em blocos. O modelo de verificação 7.0 do padrão MPEG-4 utiliza um filtro de extrapolação passa-baixas repetitivo para a codificação intra-quadro, onde os primeiros pixels a serem extrapolados (aqueles imediatamente vizinhos à região segmentada) recebem o valor do pixel do objeto vizinho horizontal, ou do vizinho vertical. Se um valor puder ser atribuído a um pixel em ambas as direções, então toma-se a média entre os dois valores. Aos demais pixels a serem extrapolados, atribui-se o valor médio dos pixels pertencentes ao objeto.

Em 1998, Moon et al propõem usar a técnica *Block Boundary Merging* (*BBM*) [71] combinada com o codificador de textura do modelo de verificação 7.0 do MPEG-4, que também utiliza a *DCT* baseada em blocos. A técnica *BBM* é aplicada após o processo de extrapolação e antes da *DCT-2D*, unindo pares de sub-blocos vizinhos pré-definidos (na horizontal, vertical ou diagonal), localizados em um mesmo macrobloco, reduzindo-se o número de blocos a serem processados através da DCT. Para análise da influência da distorção da forma, o experimento foi realizado sob condições de codificação de forma com e sem perda, havendo os autores concluído que os ganhos introduzidos pela estratégia *BBM* são praticamente indiferentes às distorções de forma. A eficiência de codificação variou segundo o tamanho e a complexidade da forma dos objetos, já que esses fatores alteram a proporção de blocos de contorno em relação ao total de blocos, bem como a razão entre o número de sub-blocos que foram unidos e o número total de sub-blocos de contorno.

Em 1999, Shen et al propuseram um novo método de extrapolação

dos blocos de contorno que, assim como a SA-DCT, apresenta a propriedade de preservação da forma [81]. Essa característica consiste em garantir que $N - L$ coeficientes (após a aplicação da DCT de N pontos sobre um segmento originalmente de tamanho L e extrapolado para N pontos no domínio espacial) sejam nulos. O algoritmo apresenta ainda a vantagem de exterminar a necessidade do processo inverso da extrapolação no decodificador, uma vez que os pixels extrapolados são relacionados aos pixels originais através de uma matriz fixa e pré-determinada.

Também em 1999, Kaup [82] resgata uma das mais promissoras técnicas de extrapolação investigadas durante o processo de normatização do padrão MPEG-4. A técnica, juntamente com a SA-DCT de Sikora, havia sido incorporada ao modelo de verificação 11.0 [17] do padrão MPEG-4 ao final de 1996. Caracteriza-se como um filtro passa-baixas, denominada *Low Pass Extrapolation* (*LPE*) e é semelhante ao filtro passa-baixas repetitivo do modelo de verificação 7.0 do MPEG-4. Consiste de três passos: (i) calcular o valor da média aritmética m dos pixels do objeto dentro da região segmentada (como no algoritmo SEA-DCT [79]), (ii) atribuir o valor m a cada pixel do bloco que esteja localizado externamente à região segmentada e (iii) aplicar a equação

$$f_{extrapolado} = \frac{f(i-1,j) + f(i,j-1) + f(i,j+1) + f(i+1,j)}{4} \qquad (2\text{-}3)$$

aos pixels imediatamente vizinhos à borda do objeto segmentado. Essa equação corresponde à média dos 4 vizinhos (V_1 a V_4) mostrados na Figura 2.5. Associado à DCT baseada em blocos, esse esquema de codificação será denominado EA-DCT-MPEG4 neste texto.

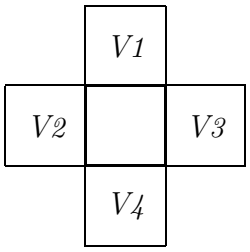

Figura 2.5: Os quatro vizinhos do pixel a ser extrapolado

Em 2000, Shen, Zeng e Liou [77] apresentaram um método simples e eficiente para aumentar a eficiência de codificação da EI-DCT, escolhendo a direção preferencial de processamento (horizontal ou vertical). Os autores

criticam [80], onde é sugerida como direção preferencial de processamento no algoritmo EI aquela que apresenta a menor variância nos comprimentos dos segmentos de objeto. Os autores afirmam que, na verdade, a opção contrária (direção que apresentar a maior variância) fornece, em média, melhores resultados [77]. O que propõem é testar as duas possibilidades (escolhendo-se cada uma das direções como preferencial) e selecionar aquela que apresentar o menor produto entre o número de bits alocados e o erro médio quadrático. A cada bloco de contorno é acrescido um bit de informação paralela, indicativo da direção escolhida como preferencial.