

República Federativa do Brasil  
Ministério do Desenvolvimento, Indústria  
e do Comércio Exterior  
Instituto Nacional da Propriedade Industrial.

(11) PI9713711-1 B1



(22) Data de Depósito: 09/12/1997  
(45) Data da Concessão: 28/06/2011  
(RPI 2112)

(51) Int.Cl.:  
H04N 7/50 2006.01

---

(54) Título: **APARELHO DE COMPRESSÃO DE BLOCOS DE PIXELS EM UM SISTEMA DE PROCESSAMENTO DE IMAGEM.**

(30) Prioridade Unionista: 12/08/1997 US 08/911525, 17/12/1996 US 60/033608

(73) Titular(es): Thomson Consumer Electronics, Inc.

(72) Inventor(es): Barth Alan Canfield, Billy Wesley Beyers, Jr., Haoping Yu, Wai-Man Lam

"APARELHO DE COMPRESSÃO DE BLOCOS DE PIXELS EM UM  
SISTEMA DE PROCESSAMENTO DE IMAGEM"

A presente invenção refere-se a um aparelho para diminuir os requisitos de memória de um processador de vídeo digital. Em particular, a invenção descreve um aparelho para comprimir com precisão as informações de pixels antes do armazenamento na memória.

O uso que economiza memória é importante no design e operação de processadores de imagem. Por exemplo, os produtos ao consumidor tais como sistemas de televisão podem usar processadores de imagem que incluem o processamento de sinais MPEG-2. O padrão de compressão de sinal MPEG (Motion Picture Experts Group) (ISO/IEC 13181-2, 10 de maio de 1994) é um padrão de processamento de imagem amplamente aceito que é particularmente atraente para uso com sistemas de transmissão via satélite, cabo e terrestre que empregam o processamento de televisão de alta definição (HDTV) entre outras formas de processamento de imagem. Os produtos que usam exibições de alta definição exigem 96 Mbits ou mais de memória para armazenar temporariamente quadros de MPEG decodificados antes da exibição. Um processador de MPEG requer esses quadros para estimativa e compensação de movimento para reconstruir imagens acuradas para exibição.

Os sistemas que reconstroem imagens a partir de elementos de imagem decodificada com MPEG (pixels ou pels) empregam Modulação Codificada de Pulso Diferencial (DPCM). No processamento de DPCM, um processador gera um valor de previsão que antecipa o próximo valor de pixel. Uma rede de

soma subtrai a previsão do valor de pixel real, o que resulta em uma diferença usada para representar os dados de vídeo. Esta diferença, conhecida como erro de previsão, geralmente é menor do que o valor de dados, de maneira tal que 5 o processamento da diferença em vez do valor de pixel original reduz os requisitos de largura de faixa do sistema. O erro de previsão pode ter um valor positivo ou negativo. Ang et al., "Video Compression Makes Big Gains" IEEE Spectrum, Outubro de 1991, descrevem um codificador e decodificador 10 de MPEG.

Os processadores de imagem que economizam memória usam menos memória para armazenar os quadros de imagem ao recodificar (recomprimir) os dados de blocos antes do armazenamento. No domínio espacial, a redução do número de bits 15 por pixel usados para armazenar os quadros de imagem afeta adversamente a qualidade da imagem se os pixels não puderem ser reconstruídos com precisão ao seu valor original. Artefatos podem ocorrer, especialmente em áreas lisas da imagem. Os processadores de imagem com redução de memória devem 20 quantificar e desquantificar com precisão o sinal decodificado com MPEG de um modo tão eficiente e econômico quanto possível.

É sabido que se tira vantagem das limitações e processamento da recepção óptica humana e dados de luminância 25 e crominância de modo diferente. As leis de otimização da compressão para cada tipo de dados para levar em conta os componentes de energia e freqüência nos dados, bem como o que o olho humano pode ver, são descritas na patente U.S.

No. 4.575.749, concedida a Acampora et al. Acampora apresenta a compressão de amplitude para reduzir o ruído em sinais de televisão antes da transmissão. Formatos de exibição tais como 4:2:2 e 4:2:0 também descrevem a compressão de dados de vídeo onde os dados de luminância e crominância foram processados de maneira diferente. Coeficientes de formato de 4:2:2 e 4:2:0 indicam que um bloco de dados de crominância contém metade ou um quarto da quantidade de informações que um bloco de dados de luminância contém. No entanto, uma vez que os dados de vídeo são recebidos em um processador de exibição, os dados são representados como dados de pixels de n bits. As técnicas de compressão conhecidas acima não se referem à compressão relativa ao processador de exibição.

No processador de exibição, os dados de luminância e crominância podem ser processados em separado, mas não com respeito à recompressão. Um exemplo de processador de exibição que processa dados de luminância e crominância de modo diferente é a conversão dos dados de razão 4:2:2 ou 4:2:0 em dados de linha de varredura, em que nem todo pixel é definido com informações de crominância. No entanto, isto não tem nada a ver com a compressão ou recompressão de dados. Até que o formato MPEG se tornasse disponível, houve pouca preocupação quanto à alocação de memória para um processador de exibição, porque não havia a necessidade de calcular um quadro de imagem a partir de vetores de movimento ou informações de composição de movimento. Com o advento do formato MPEG, os quadros múltiplos de dados de pixels não têm que

ser armazenados em memória associada com a exibição para reconstruir quadros de imagem. O pedido de depósito copendente 08/579.129 descreve a recompressão de dados de pixels de vídeo antes do armazenamento na memória de quadros, antes 5 que eles sejam recebidos pelo processador de exibição.

Mais especificamente, devido ao fato que os dados de crominância são geralmente definidos por menos pixels (com bits limitados) quando comparados aos dados de luminância (por exemplo, no formato 4:2:2 ou 4:2:0), a compressão 10 ou recompressão adicional de dados de crominância é contraindicada. A compressão ou recompressão de dados de crominância, tal como por meios tais como a quantificação, agora compromete seriamente a capacidade de reconstruir com precisão os dados de crominância originais para exibição, o que 15 resulta em qualidade reduzida de imagem. A redução de requisitos de memória para processadores de exibição tal como pode ser obtida através da recompressão de dados de pixels de luminância e crominância antes do armazenamento na memória de quadros, e a necessidade de reconstruir com precisão 20 os dados de imagem para exibição são interesses competitivos um em relação ao outro. Isto é particularmente verdadeiro no caso de um sistema de alta definição, tal como HDTV, onde os detalhes são claramente exibidos.

Os autores da presente invenção reconhecem o desejo 25 de se ter um sistema de redução de dados eficiente que empregue hardware e software mínimos que irão economizar memória e reduzir o tamanho físico do processador enquanto minimizam os artefatos introduzidos na imagem reconstruída. O

sistema descrito resolve esses problemas ao processar dados de luminância e crominância de modo diferente de acordo com os princípios da presente invenção.

Um processador de imagem de memória eficiente de acordo com a presente invenção recebe uma corrente de dados digitais de dados de vídeo com formato MPEG. Os dados MPEG são decodificados e descomprimidos e apresentados ao processador como blocos de pixels de imagem de dados de luminância e crominância. Os dados de luminância e crominância são recomprimidos a um número predeterminado de bits por blocos de pixels, sendo que a cada representação de pixels é alocado um número médio de bits para armazenamento na memória de quadros. O número médio de bits por representação de pixels é pelo menos um bit menos para os dados de crominância do que para os dados de luminância.

#### BREVE DESCRIÇÃO DOS DESENHOS

A Figura 1 é um diagrama de blocos de um processador de blocos de pixels que inclui um sistema de acordo com os princípios da presente invenção.

A Figura 2 mostra detalhes da parte de compressão do sistema da Figura 1.

A Figura 3 ilustra um formato de dados disposto em pacote adequado para uso por um sistema que inclui a presente invenção.

A Figura 4 mostra detalhes da parte de descompressão do sistema da Figura 1.

A Figura 5A mostra detalhes da parte do mapeador de quantificação da Figura 2.

A Figura 5B é uma tabela de verdade para o bloco Selecionar da Figura 5A.

As Figuras 6A, 6B e 6C são tabelas de quantificação/desquantificação de três bits, quatro bits e cinco bits, 5 respectivamente.

A Figura 7 mostra o aparelho para a produção de tabelas de desquantificação simétrica.

A Figura 8 é uma tabela que mostra os bits de quantificação suspensos.

10 As Figuras 9A, 9B e 9C representam um fluxograma de um controlador de codificação de acordo com os princípios da presente invenção.

15 A Figura 10 é um diagrama de blocos de um sistema de televisão compatível com MPEG que emprega a presente invenção.

Como uma introdução, uma modalidade exemplificadora da invenção será descrita brevemente antes de detalhar os elementos da invenção. A modalidade exemplificadora possibilita que os dados de elementos de imagem (pixels) sejam 20 comprimidos a partir de valores de oito bits a valores de quatro bits para os dados de crominância. Isto é um decréscimo de perda de 16 para 1 em resolução que, normalmente, resultaria em uma séria degradação na qualidade de imagem de vídeo. As técnicas que incorporam a invenção permitem a re-25 construção precisa de dados.

Um processador de imagem de memória eficiente determina os valores de erro de previsão de DPCM a ser quantificados para os componentes de dados de luminância e cromi-

nância de blocos de pixels. Os dados de luminância são comprimidos por uma tabela de quantificação de 64 níveis de seis bits, ao passo que os dados de crominância são comprimidos com um conjunto de tabelas de quantificação de três, 5 quatro e cinco bits personalizadas para uma faixa selecionada e acessada por esta a partir de um conjunto de faixas predeterminadas. Um pixel de referência a partir de cada blocos de pixels é comprimido de modo diferente de outros pixels para obter a precisão inicial na rede de previsão. 10 Os parâmetros de bloco são determinados, codificados e armazenados com os blocos de pixels comprimidos para facilitar a reconstrução. As tabelas de quantificação fornecem símbolos de palavras de código curtas em níveis estatisticamente prováveis de ser acessados, de modo a compensar o armazenamento 15 de parâmetros de blocos no espaço de memória fixa. Os pixels são processados individualmente para garantir a resolução máxima e uma média geral de quatro bits por pixel que inclui os parâmetros de blocos.

Antes da quantificação, os valores de erro de previsão negativos são compensados para se obter um valor positivo no âmbito do quantificador. Desse modo, o quantificador recebe valores positivos e as tabelas incluem somente pontos de decisão positivos. As tabelas simétricas permitem que os pontos médios da tabela e metade das tabelas residam 20 na memória ROM, ao passo que a outra metade é mapeada pelo circuito.

Na prática, um receptor de televisão pode incluir um decodificador MPEG. Uma rede de redução de dados quanti-

fica um sinal MPEG decodificado e descomprimido que representa blocos de imagem antes do armazenamento na memória de quadros, de modo a reconstruir os blocos conforme necessário para uma exibição da imagem. Um dispositivo de exibição exibe dados de imagem derivados da memória de quadros. Os dados recebidos e processados pela rede consistem em um sinal de vídeo de alta definição de 1920 x 1080 pixels, 4:2:2 ou 4:2:0. Os dados de luminância são partitionados em blocos de 8 x 8 pixels no domínio espacial com os dados de crominância partitionados de acordo com o formato particular.

A rede processa os dados de blocos de pixels conforme descrito acima. Para cada blocos de pixels de dados de luminância, o primeiro pixel é elevado para sete bits, de modo a eliminar o pixel menos significativo. O último pixel é quantificado com a tabela de quantificação de cinco bits provida para a faixa de 256. Todos os outros bits são quantificados com uma tabela de quantificação de seis bits. O resultado total é seis bits por recompressão de pixels.

Para os dados de crominância, a rede varre um bloco de pixels e determina a faixa, os valores de pixels mínimos e máximos para o bloco. Os valores predeterminados representativos são substituídos em lugar da faixa e dos valores de pixels mínimos, e armazenados com o valor de pixel de referência como um cabeçalho para os dados. O pixel de referência pode ser o primeiro pixel do bloco, por exemplo. Um controlador usa registros para cada bloco de pixels de crominância e seleciona uma tabela de quantificação de três, quatro ou cinco bits para processar cada pixel e manter uma

média de quatro bits por pixel após a compressão. Os símbolos de três bits em níveis selecionados das tabelas de quatro e cinco bits compensam os bits necessários para armazenar o cabeçalho. Os símbolos de três bits residem em níveis 5 onde os dados de entrada são estatisticamente mais prováveis de ocorrer, de modo a afetar, de modo mínimo, a resolução de compressão. Se os bits excessivos foram salvos ao se produzir símbolos de três bits, os símbolos de cinco bits de alta resolução são produzidos. De modo semelhante, caso bits suficientes 10 não forem salvos, uma tabela de três bits é acessada para manter uma média de quatro bits para cada pixel que inclui o cabeçalho. Os dados de luminância (luma) são reduzidos em 25% e os dados de crominância (croma) são reduzidos em 50% em relação aos blocos de pixels disponíveis a- 15 pós os dados recebidos serem descomprimidos a partir da corrente de transporte recebida. Com o sistema descrito, os dados de croma limitados por bits podem ser recomprimidos com menos bits do que os dados de luminância recomprimidos sem afetar adversamente a qualidade da imagem. O sistema 20 acima facilita a precisão durante a reconstrução do pixel.

A seguinte descrição é um exemplo dentro do contexto de um receptor de televisão de alta definição compatível com MPEG-2 para ajudar na explanação da invenção. O sistema descrito permite a compressão de comprimento fixo em 25 uma base de bloco de dados por bloco de dados para uma razão de compressão determinada ou selecionada. O armazenamento de um bloco de dados de comprimento fixo permite o acesso aleatório de blocos. O bloco de dados de comprimento fixo é

obtido com o uso de uma combinação de configurações descritas no presente documento. O sistema de acordo com os princípios da presente invenção pode ser empregado para, de modo eficiente, comprimir e descomprimir os dados de blocos de 5 qualquer fonte e não deve ser limitado a receptores de televisão.

Na Figura 1, um decodificador, por exemplo, um decodificador MPEG (não mostrado), fornece um bloco de dados de pixels decodificados MPEG para a entrada 10 de um processador com redução de memória que inclui o compressor 12. O compressor 12 inclui um previsor 18, um quantificador 20 e um combinador 22. O previsor 18 emprega princípios bem conhecidos e pode ser do tipo descrito por Jain, "Fundamentals of Digital Image Processing", Prentice-Hall, p. 484 (1989), 10 por exemplo.

O quantificador 20 fornece um bloco de pixels reduzido de dados à memória 14. Quando um processador de exibição (não mostrado) acessa o bloco de dados reduzidos na memória 14 para exibir uma imagem, o descompressor 16 reconstroi o bloco de dados original. O descompressor 16 inclui o previsor 24 e o desquantificador 26 para recuperar os dados reduzidos da memória 14 e reconstruir o bloco de dados reduzidos. O quantificador 20 e o desquantificador 26 são dispostos de acordo com os princípios da presente invenção 20 conforme será discutido. O previsor 24 é semelhante ao previsor 18.

A entrada 10 do compressor 12 recebe um bloco de pixels a partir de um decodificador MPEG, que será discutido

em associação com a Figura 10. O bloco de pixels está no domínio espacial, e contém um bloco de 8 x 8 de pixels de imagem que representa os dados de luminância (luma), por exemplo. Em um sistema que processa dados 4:2:2, os dados de 5 crominância (croma) contêm um bloco de 8 x 4 de pixels de imagem, e em um sistema que processa dados de 4:2:0, os dados de crominância (croma) contêm um bloco de 4 x 4 de pixels de imagem, por exemplo. A entrada 10 fornece os dados de blocos de pixels a uma entrada não invertida do combinador 22 e ao quantificador 20. O previsor 18 fornece dados de previsão de pixels a uma entrada invertida do combinador 22 e ao quantificador 20. O combinador 22 combina os seus sinalis a partir de entradas invertidas e não invertidas e passa a diferença ao quantificador 20. O quantificador 20 fornece 10 valores de imagem quantificados ao previsor 18 e valores de erro de previsão quantificados para a memória 14 para 15 armazenamento.

Os dados de luminância são processados de modo diferente dos dados de crominância. Cada pixel dos blocos de 20 pixels de luminância é alocado seis bits de espaço de armazenamento na memória 14 em média. O quantificador 20 seleciona um pixel de referência a partir de um bloco de pixels recebido. O pixel de referência pode ser o primeiro pixel 25 do bloco, por exemplo. O pixel de referência é um bit deslocado para direita e armazenado em uma localização predeterminada com os pixels quantificados remanescentes do bloco na memória 14. Um outro pixel do bloco é processado de modo diferente do pixel de referência e todos os outros pixels

remanescentes do bloco de pixels. Este outro pixel pode ser o último pixel do bloco de pixels, por exemplo. Ele é processado com o uso de uma tabela de quantificação de cinco bits de 26 níveis, conforme mostrado na Figura 6C. A compressão deste pixel com cinco bits compensa o processamento do pixel de referência porque a economia de um bit na posição do último pixel mantém a média de seis bits que foi comprometida ao usar sete bits para o primeiro pixel. Se o nível acessado na tabela de cinco bits contiver uma palavra de código curta, a palavra de código é zero cheia a cinco bits. Todos os outros pixels do bloco de pixels de luminância são comprimidos com o uso de uma tabela de quantificação de seis bits de 64 níveis. A tabela de quantificação é projetada para aceitar somente valores de erro de previsão DPCM positivos. Os detalhes do design desta tabela são os mesmos para todas as tabelas de quantificação neste sistema, e serão discutidos posteriormente. Também será discutida posteriormente a compensação de valores de erro de previsão negativos para garantir valores de entrada positivos nas tabelas de quantificação, que é o mesmo para dados tanto de luma quanto de croma.

Os dados de croma são processados e comprimidos de uma maneira diferente dos dados de luma. A Figura 2 ilustra o quantificador 2 com mais detalhes uma vez que ele pertence aos dados de croma. As mesmas referências numéricas identificam elementos comuns nas Figuras 1 e 2. Especificamente, o quantificador 20 inclui o processador de erro de previsão 27, o mapeador de quantificação 28, o controlador de codifi-

cação 29, o processador de faixa mín-máx (MMRP) 30, o primeiro processador de pixels 31, e o multiplexador 32. A entrada 10 fornece os dados de pixels de bloco ao MMRP 30, que varre o bloco de pixels e determina o valor mínimo de pixels, o valor máximo de pixels e a faixa para o bloco. O MMRP 30 seleciona uma faixa predeterminada a partir de um conjunto de faixas predeterminadas como uma função da faixa real, e troca a faixa predeterminada selecionada pela faixa real para uso subseqüente na rede. O MMRP 30 comprime os valores de parâmetros de bloco de faixa mínimos, máximos e predeterminados, e os transfere para o multiplexador 32.

O valor de pixel mínimo e a faixa também são transferidos para o primeiro processador de pixels 31, e a faixa predeterminada é transferida para o processador de erro de previsão 27, conforme será discutido. O processador de erro de previsão 27 recebe os dados de erro de previsão do combinador 22 e compensa os valores negativos com a faixa predeterminada selecionada. O mapeador de quantificação 28 recebe os valores de erro de previsão compensados e não compensados do processador de erro de previsão 27. Esses valores são quantificados e enviados ao multiplexador 32. O mapeador de quantificação 28 também envia os valores de erro de previsão quantificados ao previsor 18, os quais o previsor 18 usa para calcular os dados de previsão. O multiplexador 32 envia os parâmetros de blocos e os dados quantificados à memória 14 sob cronometragem e controle que serão discutidos posteriormente. Os parâmetros de bloco representam dados de cabeçalho (overhead) que são armazenados na me-

mória 14 dentro de um campo de parâmetro associado com o bloco de pixels quantificado. O campo de parâmetro e os dados quantificados formam em conjunto um pacote que consolida todas as informações necessárias pelo descompressor 16 para 5 acessar as tabelas de desquantificação apropriadas e reconstruir o bloco de pixels. O controlador de codificação 29 supervisiona a transferência de parâmetros de bloco e dados comprimidos, bem como a seleção de tabelas de quantificação para os blocos de pixels individuais, conforme será discutido. 10

O primeiro processador de pixels 31 recebe o bloco de pixels da entrada 10 e identifica um valor de pixel de referência predeterminado. O valor de pixel mínimo de bloco recebido do MMRP 30 facilita a compressão do pixel de referência 15 independentemente de outros pixels de bloco. O pixel de referência comprimido é representado com bits suficientes para o desquantificador 26 reconstruir o seu valor original de uma maneira sem perda ou quase sem perda. O primeiro processador de pixels 31 passa o valor de pixel de referência 20 comprimido como um parâmetro de bloco ao multiplexador 32, o qual transfere os parâmetros de bloco, inclusive o valor de pixel de referência, e os dados quantificados à memória 14. O desquantificador 26 usa o pixel de referência como 25 um valor de previsão para os pixels de bloco quantificados durante a descompressão do pixel. Uma vez que o primeiro valor (o valor do pixel de referência) usado na rede de previsão durante a descompressão é independente, um determinado bloco de pixels pode ser descomprimido sem informações

de outros blocos de pixels. Este valor também é preciso, o que elimina um erro de previsão que se propaga dos dados reconstruídos.

O pixel de referência é comprimido com o uso do valor mínimo do bloco de pixels como um previsor para a derivação do valor comprimido. O valor mínimo é subtraído do valor de referência, e a diferença é dividida por dois. O resultado é armazenado na memória 14 com um bit menos do que o necessário para uma representação binária da faixa predeterminedada. A faixa predeterminada define o número de bits usados para armazenar o valor de pixel de referência comprimido porque, ao usar valores de pixels de bloco como previsores para outros valores no mesmo bloco de pixels, a diferença entre quaisquer dois valores de pixels de bloco, tal como os valores de pixels de referência e mínimo, incidirá no domínio da faixa. O valor de referência comprimido usa um bit menos do que o necessário para representar a faixa porque a diferença é dividida por dois, o que reduz o número de bits necessários para uma representação binária por um bit.

O quantificador 20 e o desquantificador 26 acessam as tabelas de quantificação e desquantificação, respectivamente, que são otimizadas para cada bloco. As tabelas de quantificação e desquantificação incluem os valores baseados em uma faixa aproximada de blocos de pixels. O quantificador de faixa mín-máx MMRP 30 recebe um bloco de dados de entrada e o varre, para determinar o valor de pixel mínimo e o valor de pixel máximo. O MMRP 30 então subtrai o valor de

pixel mínimo do valor de pixel máximo e adiciona um (máximo+1) para calcular a faixa para o bloco de pixels.

O quantificador 20 compara a faixa calculada a um conjunto de faixas predeterminadas, pelo menos uma das quais 5 é maior ou igual à faixa calculada, seleciona uma faixa predeterminada, e acessa as tabelas de quantificação derivadas da faixa predeterminada selecionada. A faixa predeterminada é selecionada por meio de uma análise de ajuste melhor que identifica a faixa predeterminada que é o menor valor do 10 conjunto que é maior ou igual à faixa calculada real. As tabelas de quantificação e desquantificação são personalizadas para incluir valores dentro do domínio da faixa predeterminada selecionada e, por conseguinte, incluem valores de toda a faixa real. O quantificador 20 emprega processamento 15 DPCM e produz valores de diferença que são erros de previsão. Esses erros de previsão ficam no domínio da faixa real se os valores de pixels fornecidos ao previsor 18 forem oriundos do mesmo bloco de pixels que o pixel para o qual o previsor 18 está atualmente gerando um valor de previsão. O 20 compressor 12 segue e mantém este parâmetro. A faixa real de um determinado bloco de pixels é normalmente significativamente inferior a 256 (o valor máximo de um valor de pixel de 8 bits), e os níveis da tabela derivados da faixa predeterminada produzem uma resolução melhor do que os níveis da 25 tabela derivados de 256, porque a faixa predeterminada selecionada está geralmente próxima em valor da faixa real. Por conseguinte, a precisão e a eficiência do sistema aumentam com a personalização dos níveis da tabela à faixa.

Para reconstruir os dados do bloco de entrada, o desquantificador 26 deve saber qual faixa predeterminada o quantificador 20 usou para acessar a tabela de quantificação empregada ao quantificar o bloco de pixels. As representações da faixa e outros parâmetros de blocos de pixels são armazenados na memória 14 dentro de um campo de parâmetro com o bloco de pixels quantificado. Ao armazenar uma representação de parâmetro de bloco na memória 14 com o bloco de pixels quantificado, o descompressor 16 pode acessar a tabela de desquantificação adequada e reconstruir o bloco de pixels de modo eficiente e preciso. Outros parâmetros de blocos de pixels incluídos no campo de parâmetros podem ser, por exemplo, o valor de blocos de pixels mínimo ou um valor de blocos de pixels de referência. A Figura 3 ilustra uma configuração possível de um campo de parâmetros e dados comprimidos. O campo de parâmetros consiste nos parâmetros de bloco contidos na caixa tracejada na Figura 3. Nesta modalidade, um campo de parâmetros está disposto como um cabeçalho de um pacote de dados que contém uma carga útil de dados comprimidos.

Para maximizar a redução da memória de quadros sem degradar de maneira significativa a imagem exibida, as informações suspensas representadas pelos parâmetros de bloco no campo de parâmetros são armazenadas na memória 14. Todo bit usado para armazenar o campo de parâmetros diminui a memória disponível para o armazenamento de pixels quantificados. Por conseguinte, um sistema que usa a presente invenção diminui o número de bits necessários para armazenar dois

dos parâmetros de blocos, ou seja, a faixa e o valor mínimo, de oito bits a três bits para cada parâmetro na maioria dos casos. Este processo funciona tal como segue.

A faixa real é comparada com um conjunto de faixas predeterminadas para determinar um melhor ajuste. A faixa predeterminada torna-se o valor usado para representar a faixa para os blocos de pixels então sendo processados. A faixa predeterminada é maior do que a faixa real para garantir que todos os valores de pixels no bloco de pixels sejam representados. O conjunto de faixas predeterminadas inclui sete valores, que são 16, 32, 64, 96, 128, 192 e 256. Uma vez que o conjunto está disponível tanto ao quantificador 20 quanto ao desquantificador 26, a faixa predeterminada pode ser representada no campo de parâmetros por um valor de índice. O índice requer somente três bits para uma representação binária porque somente há sete faixas predeterminadas para representar.

O sistema lida com o valor de pixel mínimo de uma maneira semelhante. Para cinco das sete faixas predeterminadas, o sistema acessa um conjunto predeterminado de oito valores de pixels mínimos únicos à faixa predeterminada selecionada. O quantificador 20 compara o valor de pixel mínimo real ao conjunto predeterminado e seleciona o maior valor mínimo predeterminado que é menor ou igual ao valor mínimo real. O mínimo predeterminado torna-se então o valor usado para representar o pixel mínimo para o bloco de pixels sendo processado. O conjunto fica disponível tanto para o quantificador 20 quanto para o desquantificador 26, de ma-

neira tal que o mínimo predeterminado pode ser representado no campo de parâmetros por um valor de índice. Este índice também requer três bits para uma representação binária porque há somente oito valores de pixels mínimos predeterminados para representar.

O conjunto de oito valores de pixels mínimos predeterminados para cinco das sete faixas é definido pela equação (1a) abaixo. As cinco faixas às quais a equação (1) se aplica são 32, 64, 96, 128 e 192. A equação propicia uma etapa linear constante para cada faixa mínima que começa com zero.

A equação (1) abaixo seleciona o valor de pixel mínimo predeterminado a partir do conjunto  $Q_{min}(Rs, i)$  (equação (1a)) que é substituído pelo valor de blocos de pixels mínimo real.  $MAX_i\{f(x)\}$  indica que o valor máximo de  $i$  que satisfaz a condição nas chaves deve ser usado para gerar  $Q_{min}$ .

$$Q_{min} = MAX_i\{Q_{min}(Rs, i) | Q_{min}(Rs, i) \leq X_{min}; 0 \leq i \leq 7\}, \quad (1)$$

onde:

$$20 \quad Q_{min}(Rs, i) = INT\{i((256-Rs)/7)\}; \quad 0 \leq i \leq 7. \quad (1a)$$

Nessas equações,  $i$  é o valor de índice representado por três bits no campo de parâmetros suspensos.  $INT\{f(x)\}$  indica que somente a parte do número inteiro do valor resultante é usada. A expressão  $f(x)$  entre chaves é representante de qualquer expressão, tal como aquela na equação (1), sob a qual a função  $INT$  opera. Para a faixa predeterminada de 256, nenhum valor mínimo é armazenado porque o valor mínimo para 256 é zero (0) para uma palavra de oito bits. Para a

faixa predeterminada de 16, o valor mínimo de oito bits original é usado, porque a resolução para esta faixa é pequena em relação aos valores mínimos disponíveis a partir da equação (1). O uso da equação (1) para gerar um valor mínimo para a faixa de 16 pode fazer com que os valores de pixels reais caiam fora dos dados reconstruídos na reconstrução. O valor mínimo é um deslocamento que representa a distância entre zero e o valor mínimo de pixels de bloco.

A equação (1) pode selecionar uma faixa predeterminada que não é suficiente para cobrir os valores de blocos de pixels reais quando os blocos de pixels quantificados são reconstruídos, porque os valores mínimos predeterminados são menores do que o valor mínimo real. Por exemplo, se em um determinado bloco de pixels o valor de pixel mínimo for 100 e o valor de pixel máximo for 140, então a faixa predeterminada selecionada é 64. O valor de pixel mínimo predeterminado selecionado resultante da equação (1) é 82. O resultado da adição do mínimo selecionado à faixa selecionada é 146, que é maior do que o valor de pixel máximo real. Por conseguinte, todos os valores do bloco de pixels serão representados pelos valores predeterminados selecionados. No entanto, se o valor de blocos de pixels máximo for, em vez disso, 160, os valores predeterminados selecionados permanecerão os mesmos, mas não representam completamente o domínio do bloco de pixels ( $160 > 146$ ). Neste caso, a próxima faixa predeterminada mais alta de 96 é selecionada e um novo valor mínimo predeterminado selecionado é 91. A soma de 91 e a faixa predeterminada de 96 é 187, que é maior do que o valor

de blocos de pixels máximo real de 160. Por conseguinte, as tabelas de quantificação e desquantificação selecionadas a partir desta faixa irão fornecer níveis para todos os pixels no bloco. O quantificador 28 executa a análise descrita a-cima para determinar se a primeira seleção de faixa predeterminada e valores de pixels mínimos é válida, ou se a próxima faixa predeterminada maior é necessária.

Conforme declarado anteriormente, se a rede de previsão deriva seus valores de previsão de valores de pixels no mesmo bloco, então a diferença ( $E$ ) entre um valor de pixel real e o valor de pixel previsto estará dentro dos seguintes limites:

$$-Faixa < E < Faixa, \text{ onde} \quad (2)$$

$$Faixa = X_{\max} - X_{\min} + 1 \quad (3)$$

Na equação (2),  $E$  é o erro de previsão. Na equação (3),  $X_{\max}$  e  $X_{\min}$  são os valores de pixels de bloco máximo e mínimo, respectivamente. Por conseguinte, a faixa de dados de pixels do bloco define os valores que as tabelas de quantificação e desquantificação receberão, e os limites para os quais as tabelas devem prover para aquele bloco particular. Se a faixa for menor do que o valor máximo do tamanho da palavra (256 para uma palavra de 8 bits), então a resolução das tabelas de quantificação e desquantificação pode ser aumentada.

Os dados tanto de luma quanto de croma são processados com erros de previsão negativos compensados. Por conseguinte, as tabelas de quantificação e desquantificação para dados de luma e croma são projetadas para aceitar somente

os valores de entrada positivos. As tabelas de quantificação e desquantificação que empregam valores de erro de previsão negativos compensados têm o dobro da resolução que as tabelas projetadas somente para a faixa do bloco de pixels.

5 A resolução é duplicada porque as tabelas precisam somente cobrir valores de zero ao valor de faixa positivo em vez de todos os valores entre a faixa positiva e negativa. As Figuras 6A, 6B e 6C mostram tabelas de três bits, quatro bits e cinco bits, respectivamente, para a faixa predeterminada de 64. Antes da quantificação, o processador de erro de previsão 27 (Figura 2) detecta se o erro de previsão do combinador 22 é positivo ou negativo. Se o valor for positivo, ele passa inalterado para o mapeador de quantificação 28. Se o valor for negativo, o processador de erro de previsão 27 adiciona a faixa predeterminada ao valor de erro de previsão negativo antes de o valor ser passado para o mapeador de quantificação 28. Uma vez que um valor de erro de previsão negativo esteja dentro do domínio do valor de faixa negativo, a adição do valor de faixa positivo ao valor de erro de previsão negativo resulta em um valor de erro compensado. Este valor de erro compensado é positivo (maior do que zero) e é inferior ao valor de faixa positivo. O mapeador de quantificação 28 recebe os valores de erro de previsão tanto compensados quanto não compensados e os quantifica com uma tabela de quantificação adaptada ao domínio da faixa predeterminada positiva. Os valores de erro quantificados são passados ao multiplexador 32 e então armazenados na memória 14 sob o controle de um controlador de sistema (não mostra-

do). Uma vez que a tabela somente quantifica valores de zero à faixa -1 em vez de fazê-lo do valor de faixa negativo para o valor de faixa positivo, a resolução da tabela é duplicada.

5 A Figura 4 é um diagrama de blocos do desquantificador 26 da Figura 1. Sob o controle de um microprocessador de sistema, o desmultiplexador 34 recebe um pacote de dados que contém um campo de parâmetros e dados quantificados. O desmultiplexador 34 envia o índice do valor de pixel mínimo 10 e o índice da faixa predeterminada ao decodificador de faixa-mín-máx (MMRD) 38. O desmultiplexador 34 envia o valor do primeiro pixel comprimido ao primeiro decodificador de pixels 37, que também recebe a faixa reconstruída predeterminada e os valores de pixels mínimos do MMRD 38. O primeiro 15 decodificador de pixels 37 usa esses três valores para reconstruir o pixel de referência e enviá-lo ao previsor 24. Na desquantificação, o desmultiplexador 34 envia os valores quantificados ao mapeador do desquantificador 36, o qual desquantifica os valores de erro de previsão e os passa ao adicionador 39. O adicionador 39 adiciona o valor previsto 20 ao valor de erro desquantificado e passa o resultado ao processador de erro de previsão 35, o qual compara o resultado com o valor de bloco de pixels máximo reconstruído. Se o valor de erro foi compensado para ter um valor negativo a 25 um valor positivo antes da quantificação, o resultado será maior do que o valor de pixel máximo reconstruído. Caso contrário, o resultado será menor ou igual ao valor de pixel máximo reconstruído. Se o processador de erro de previsão

35 determinar que o valor de erro foi compensado, o valor de faixa predeterminado é subtraído do resultado, de modo a corrigir para a compensação introduzida no lado da quantificação da rede. O processador de erro de previsão 35 e o 5 primeiro decodificador de pixels 37 passam os dados reconstruídos, inclusive o pixel de referência, na devida ordem a uma rede de saída (não mostrada).

Os valores disponíveis para o desquantificador 26 são valores quantificados e/ou codificados. O valor de pixel 10 mínimo quantificado reconstruído ( $Q_{min}$ ) deve ser menor ou igual ao valor de pixel mínimo real, e o valor de pixel máximo reconstruído ( $Q_{max}$ ) e o valor da faixa quantificado reconstruído devem ser maiores ou iguais aos seus valores reais. O MMRP 30 garante que esses requisitos sejam satisfeitos, conforme anteriormente discutido. Uma vez que qualquer 15 valor de pixel deve ser maior ou igual a  $Q_{min}$ , a adição da faixa predeterminada a qualquer valor de pixel reconstruído que inclui a compensação geralmente resulta em um valor maior do que  $Q_{max}$  em pelo menos um.

20 No entanto, o ruído de quantificação  $N_q$  pode causar uma determinação incorreta do fato se o quantificador detectou um valor de erro de previsão negativo e compensou o valor. O ruído de quantificação é a diferença entre o valor de pixel real e o valor reconstruído causado pela resolução 25 nas tabelas de quantificação de perda. O processador de erro de previsão 35 adiciona o nível reconstruído à faixa predeterminada e compara o resultado com  $Q_{max}$ . Se o resultado for maior do que  $Q_{max}$ , a faixa predeterminada é subtraída do

resultado para se obter o valor de pixel reconstruído correto. Mas, se  $N_q$  for positivo, ele pode fazer o resultado ser maior do que  $Q_{max}$ , e o processador de erro de previsão 35 identificaria falsamente um erro de previsão compensado. De modo semelhante, se  $N_q$  for negativo, ele pode fazer com que o resultado seja inferior a  $Q_{max}$ , e o processador de erro de previsão 35 identificaria falsamente um erro de previsão não compensado.

A Figura 5A ilustra quanto o mapeador do quantificador 28 (Figura 2) garante que sua saída não será mal interpretada devido ao ruído de quantificação. O quantificador 80 fornece três saídas para cada valor de pixel quantificado. Os três valores são o melhor nível de reconstrução para o ponto de decisão da tabela de quantificação (I), e o nível de reconstrução em cada lado do melhor nível (I+1, I-1). O combinador 84 calcula o valor de pixel reconstruído para o melhor valor de reconstrução e o resultado é comparado com  $Q_{max}$  pelo combinador 86. Se o erro de previsão foi compensado ( $S_2$  é negativo) e o resultado do combinador 86 for inferior a  $Q_{max}$  ( $S_1$  é negativo), é possível que na reconstrução o processador de erro de previsão 35 determine incorretamente que o valor de erro de previsão não foi compensado. Para acabar com esse problema, a palavra de código correspondente ao próximo nível de reconstrução maior para o erro de previsão é enviada ao multiplexador 32. Se o erro de previsão não foi compensado ( $S_2$  é positivo) e o resultado do combinador 86 for maior do que  $Q_{max}$  ( $S_1$  é positivo), é possível que na reconstrução o processador de erro de previsão

são 35 determine incorretamente que o valor de erro de previsão desquantificado foi compensado. Para acabar com esse problema, a palavra de código correspondente ao próximo nível de reconstrução menor para o erro de previsão é enviada 5 ao multiplexador 32. Em todos os outros casos, o melhor nível é selecionado e enviado ao multiplexador 32.

Quando o primeiro ou o último nível em uma tabela de quantificação for o melhor, somente o próximo nível de quantificação maior ou menor é provido com o melhor nível. 10 A Figura 5B mostra uma tabela de verdade que ilustra as escolhas do quantificador 80 disponíveis para saída pelo mapeador de quantificação 28, e quando a unidade selecionada 82 usa cada uma das escolhas. Uma vez que o ruído de quantificação pode fazer com que a análise de correção de erro de 15 compensação esteja incorreta, a escolha de um valor de ruído de quantificação com um sinal oposto não vai influenciar a relação entre  $Q_{max}$  e o valor do pixel reconstruído.

Como o valor absoluto de  $Nq$  geralmente não é muito grande, o marcador de quantificação 28 vai normalmente escolher o melhor nível de quantificação. Quando o marcador de quantificação 28 escolhe o próximo nível maior ou menor, a escolha vai induzir um erro adicionado no pixel reconstruído. No entanto, o erro é minimizado com a seleção do nível mais próximo que vai corrigir o problema em uma tabela com 25 resolução que é muito melhor do que as tabelas de quantificação DPCM conhecidas. Geralmente, esta correção não causa uma degradação perceptível na qualidade da imagem exibida.

A resolução da quantificação é normalmente aumentada mais do que apenas o fator de dois que ocorre a partir da compensação de erros de previsão negativos para produzir valores positivos. A seleção de uma faixa predeterminada 5 também leva a uma resolução de quantificação aumentada. Por exemplo, se para um determinado bloco de pixels a faixa predeterminada selecionada for 16, então a tabela de quatro bits vai quantificar com exatidão o valor de erro de previsão de uma maneira sem perdas. A resolução aumenta em um 10 fator de 16 de uma faixa de 256 (para 8 bits) a 16 (256/16). Ao requerer somente valores positivos na tabela de quantificação/desquantificação para o mesmo valor de faixa positivo, a resolução aumenta em outro fator de 2 de um fator total de 15 32. Este processo pode ser usado para calcular o aumento na resolução para as tabelas de quantificação/desquantificação derivadas de qualquer um dos valores de faixa predeterminada.

Quando o MMRP 30 (Figura 2) varre um bloco de pixels de croma e seleciona uma faixa predeterminada a partir 20 do conjunto de sete, os dados podem ser graduados para reduzir as tabelas de quantificação reais necessárias para comprimir os dados. Para as faixas predeterminadas de 32, 192 e 256, há um conjunto de três tabelas personalizadas. Estas 25 são as tabelas de três, quatro e cinco bits conforme discutido anteriormente. Os dados que se ajustam a três das outras faixas predeterminadas são dados graduados para cima em um fator de dois. Os blocos de pixels que se ajustam à faixa predeterminada de 32 são graduados para utilizar as tabe-

las de quantificação acessadas pela faixa de 64. Para os blocos de pixels que se ajustam à faixa predeterminada de 96, as tabelas de quantificação para 192 são utilizadas. De modo semelhante, para os blocos de pixels que se ajustam à 5 faixa predeterminada de 128, as tabelas de quantificação para 256 são utilizadas. A graduação dos dados reduz a complexidade de hardware e software, e reduz a quantidade de memória exclusiva para leitura (ROM) necessária na rede. Na reconstrução, os blocos de pixels graduados são divididos 10 por dois após a desquantificação para manter a precisão dos dados.

Todas as tabelas de quantificação para dados tanto luma quanto croma são construídos de modo a ser simétricos ao redor de seu ponto médio. As Figuras 6A, 6B e 6C mostram 15 as tabelas de quantificação de três, quatro e cinco bits para a faixa predeterminada de 64. A simetria permite que somente metade de cada tabela seja armazenada na memória ROM, ao passo que um simples circuito de hardware implementa a outra metade de cada tabela. Isto reduz o tamanho da memória ROM de modo a reduzir os custos de produção.

As tabelas de quantificação são projetadas para um conjunto simples de relações, dadas abaixo, que criam uma simetria ao redor do ponto médio da tabela. Nessas relações,  $I$  é o índice de nível de quantificação;  $D_I$  é o Inésimo 25 ponto de decisão;  $Q_I$  é o Inésimo nível de reconstrução;  $M$  é o número total de níveis em uma tabela; e  $R$  é a faixa de quantificação. Um ponto de decisão é o valor no qual uma entrada de valor de pixel em uma tabela de quantificação se

move de um nível a outro dentro da tabela. As relações são tal como segue:

M é um número par; (4)

$$D_I + D_{M-1-I} = R_d - 1 \text{ para } 0 \leq I \leq (M/2) - 1; \quad (5)$$

5  $Q_0 = 0; \quad (6)$

$$Q_{M/2} = R_d/2; \quad (7)$$

$$Q_I + Q_{M-I} = R_d \text{ para } 1 \leq I \leq M/2 \quad (8)$$

As tabelas nas Figuras 6A, 6B e 6C se conformam a estas relações tal como todas as tabelas de quantificação 10 usadas na rede. Se as tabelas aceitassem valores de erro de previsão que não fossem compensados, em outras palavras, valores tanto positivos quanto negativos, então M seria um número ímpar.

Os símbolos de palavra de código de saída das tabelas 15 também têm relações para garantir a simetria, de modo a permitir vantajosamente que somente metade das tabelas seja armazenada na memória ROM. Todos os símbolos nessas relações são os mesmos nas relações acima. As únicas adições são  $C_I$ , que é a palavra de código para o Inésimo nível e  $n$ , 20 que é o número de bits na palavra de código. Essas relações são tal como segue:

$$C_1 = 0 \text{ e } C_{M-1} = 1 \text{ para palavras código curtas; } (9)$$

$$C_0 = (2^n) - 2; \quad (10)$$

$$C_{M/2} = (2^n) - 1; \quad (11)$$

25  $C_I = 2I \text{ para } 2 \leq I \leq (M/2) - 1; \quad (12)$

$$C_{(M/2)+I} = C_{(M/2)-I} + 1 \text{ para } 1 \leq I \leq (M/2) - 1. \quad (13)$$

As relações (9) a (13) são representadas nas tabelas na forma binária. Se as tabelas aceitassem valores de erro de

previsão que não fossem compensados, em outras palavras, valores tanto positivos quanto negativos, então também haveria um número ímpar de palavras código de três bits.

A relação (9) define a colocação estatisticamente ótima para as palavras código de três bits para as tabelas de quantificação que recebem valores de erro de previsão DPCM. Ao colocar as palavras código curtas em níveis de reconstrução que têm mais probabilidade de ser acessados, o sistema é otimizado para economizar bits de armazenamento para outros propósitos. O uso de palavras código curtas diminui o número de níveis disponíveis em uma tabela de quantificação, mas as economias de bits excedem em valor a resolução perdida porque geralmente bits suficientes são economizados na memória para acomodar as informações suspensas e manter a média de quatro bits por palavra de código.

Devido ao fato que as tabelas ficam simetricamente ao redor de seu ponto médio, somente metade dos níveis de tabela, inclusive o nível do ponto médio, precisa ser realmente armazenada na memória. Os níveis remanescentes podem ser mapeados por circuito. A Figura 7 mostra uma implementação de hardware simples para o mapeamento dos valores de reconstrução de tabela remanescentes na desquantificação, e saída de todos os valores de pixels reconstruídos de oito bits para ambas as metades das tabelas.

O controlador de codificação 29 (Figura 2) e o controlador de decodificação 33 (Figura 4) desempenham mutuamente operações similares, porém inversas para os blocos de pixels de croma. Ambos incluem quatro registros que contam

o número de pixels processados para cada bloco, e o número de bits conservados ou necessários para as informações suspensas. Um registro, o registro de faixa, é um registro de sinalizador que identifica qual faixa predeterminada representa o bloco de pixels atualmente sendo processado. Na utilização dos registros, os controladores 29 e 33 selecionam para cada pixel processado a tabela de quantificação de três, quatro ou cinco bits e garantem que o bloco de pixels, inclusive toda as informações suspensas, seja comprimido a um tamanho predeterminado para armazenamento na memória 14, e então descomprimido e reconstruído ao bloco de pixels original.

As informações suspensas incluídas para dados de croma requerem um número predeterminado de bits dependendo dos parâmetros de bloco a ser armazenados. A Figura 8 mostra o número de bits suspensos necessários para cada parâmetro de bloco para cada faixa predeterminada. Cada pixel, inclusive o pixel de referência, tem uma média de quatro bits reservados na memória 14. O controlador 29 compensa quatro bits menos do que o número total de bits usados para o cabeçalho. O número de bits necessários como compensação para bits de cabeçalho (overhead) para cada faixa predeterminada é mostrado na última fileira da Figura 8.

O objetivo principal do controlador 29 é codificar cada pixel com a tabela de quatro ou cinco bits e somente usar a tabela de três bits quando necessário para garantir que todos os pixels de bloco se ajustem no espaço reservado. As palavras código curtas de três bits nas tabelas de quatro

e cinco bits propiciam a melhor oportunidade para atingir este objetivo. Uma vez que as palavras de código curtas são estatisticamente colocadas nas tabelas em níveis mais prováveis de ser acessados para os dados DPCM, cada bloco será 5 normalmente comprimido sem usar a tabela de quantificação de três bits. Além disso, geralmente vários pixels dentro de qualquer bloco de pixels serão quantificados com palavras de código de cinco bits, de modo a aumentar a resolução e a exibir qualidade. No entanto, se os blocos de pixels não 10 acessarem as palavras código nas tabelas de quantificação de quatro e cinco bits normalmente suficientes para compensar o número necessário de bits suspensos, o controlador 29 acessa a tabela de quantificação de três bits. O controlador 29 identifica os últimos N bits em cada bloco de pixels como 15 pixels de baixa prioridade (LPP), onde N é o número de bits suspensos compensado para esse bloco de pixels. Com base nos contadores que identificam quando um LPP está sendo processado e quantos bits suspensos permanecem não compensados, o controlador 29 seleciona a tabela de quantificação de três 20 bits para os LPPs. O controlador 29 não seleciona a tabela de quantificação de três bits até que o número de pixels que permanecem para ser comprimidos se iguale ao número de bits suspensos que permanecem não compensados.

As Figuras 9A, 9B e 9C mostram um fluxograma dos 25 controladores 29 e 33. Ambos os controladores operam da mesma maneira, e executam as mesmas etapas para comprimir ou descomprimir um valor de pixel. Para simplificar a discussão,

são dos controladores 29 e 33, somente o controlador de compressão 29 será explanado.

No Início, quatro registros são inicializados no começo de cada bloco de pixels de croma. O registro de faixa é codificado de acordo com a faixa predeterminada para o bloco de pixels atual. O registro auxiliar (overhead register) é configurado para o número de bits iniciais (overhead bits) os quais devem ser compensados pelo controlador 29, conforme mostrado na Figura 8. Este registro é diminuído em uma unidade toda vez que um LPP é processado. O registro de economia de bits é inicializado ao valor negativo do registro auxiliar, e é incrementado cada vez que uma palavra de código é usada. O registro de contagem de pixels é inicializado ao valor negativo do número de pixels no bloco de pixels atual. O registro é aumentado cada vez que um pixel é processado. A contagem de pixels é usada para identificar se o pixel sendo processado é um LPP.

No Início na Figura 9A, um valor de pixel de oito bits passa para a etapa 100 e identifica se o pixel é um pixel de baixa prioridade (LPP). Em caso positivo, a etapa 102 adiciona economia de bits e cabeçalho e compara o resultado com zero. Esta é uma etapa significativa porque se bits suficientes não foram economizados neste ponto, a tabela de quantificação de três bits é acessada e uma palavra de código de três bits é usada para comprimir o pixel na etapa 104. Uma vez que o pixel atual é um LPP, todos os pixels precedentes processados podem não ter sido comprimidos com as palavras de código curtas de três bits na tabela de qua-

tro bits um número suficiente de vezes para compensar todos os bits suspensos no bloco de pixels. Por conseguinte, se os valores de registro de economia de bit + cabeçalho > 0, um bit deve ser economizado aqui e a tabela de três bits de 5 baixa resolução é usada para comprimir o pixel. Ao mesmo tempo, os registros de economia de bits e a contagem de pixels são incrementados, e o registro suspenso é diminuído. Neste ponto, o pixel é comprimido e o próximo pixel é processado a partir do Início.

10 Se, na etapa 102, o resultado for maior do que zero, a etapa 106 determina se a economia de bits é maior do que zero. Em caso negativo, há bits apenas suficientes salvos para compensar o cabeçalho para esse ponto, e há o mesmo número de LPPs remanescentes que os bits suspensos não compensados. Então a etapa 104 aumenta a economia de bit e a contagem de pixels, diminui o cabeçalho e as saídas da palavra de código de três bits da tabela de três bits. Se a economia de bits for maior do que zero na etapa 106, a etapa 15 108 verifica a única circunstância de economia de bit = 0 e faixa = 16. Se isto ocorrer, a etapa 110 acessa a tabela de quantificação de quatro bits de nível dezesseis, sem palavra de código curta porque não há necessidade de economizar um bit para este pixel. A contagem de pixels é aumentada, o cabeçalho é diminuído, e o próximo pixel é recuperado para 20 compressão. Se o resultado da etapa 108 for não, a tabela de quatro bits para a faixa predeterminada atual é acessada na etapa 112. A etapa 114 verifica se o valor de pixel incide em um nível de palavra de código curta da tabela de 25

quatro bits. Em caso positivo, a etapa 116 aumenta a economia de bits e a contagem de pixels, diminui o cabeçalho (overhead) e fornece a palavra de código de três bits. Em caso negativo, a etapa 118 aumenta a contagem de pixels, diminui o cabeçalho (overhead) e fornece a palavra de código de quatro bits. Após ambas as etapas 116 e 118, o próximo pixel é recuperado para processamento que começa no Início.

Voltando à Etapa 106, e lembrando que o pixel foi determinado como sendo um LPP na etapa 100, se a economia de bits for maior do que zero, então o processo continua na Figura 9B. Uma vez que a economia de bits é maior do que zero, houve um número mais do que suficiente de bits economizados para esse ponto no bloco de pixels. Por conseguinte, a tabela de cinco bits de alta resolução é usada na etapa 120. A tabela de cinco bits contém palavras de código curtas, e a etapa 122 determina se o valor de pixel incide em um nível de palavra de código curta da tabela. Em caso negativo, a etapa 132 aumenta a contagem de pixels, diminui a economia de bits e do cabeçalho e fornece uma palavra de código de cinco bits. Em caso positivo, a etapa 124 adiciona economias de bits e a contagem de pixels para determinar se muitos bits foram economizados. Se o número de bits economizados ficar muito grande, a sincronização do sistema não é mantida. Para impedir isto e a necessidade de um buffer para conter os dados comprimidos até que o sistema se recupere, palavras de código de três bits podem ser preenchidas com zero. O resultado da etapa 124 determina que caminho a compressão do pixel vai prosseguir. Se o resultado for me-

nor do que zero, então não foram salvos bits suficientes e nenhum preenchimento com zero ocorre na etapa 126. A economia de bit e a contagem de pixels são incrementadas, o cabeçalho é diminuído e a palavra de código de três bits é fornecida. Se o resultado for igual a zero, então a palavra de código de três bits vai economizar um para muitos bits. Por conseguinte, na etapa 128, a palavra de código é preenchida com um zero, a contagem de pixels é aumentada, o cabeçalho é diminuído e a palavra de código de quatro bits preenchida é fornecida. Se o resultado na etapa 124 for igual a 1, então a palavra de código de três bits vai economizar dois para muitos bits. Por conseguinte, na etapa 130, a palavra de código é preenchida com dois zeros, a contagem de pixels é aumentada, a economia de bits e os cabeçalhos são diminuídos, e a palavra de código de cinco bits preenchida é fornecida. Após as etapas 126, 128, 130 e 132, a compressão do pixel é completa e o próximo pixel é recuperado para processamento no Início.

Se na etapa 100 o pixel for identificado como não sendo um LPP, o processo vai para a etapa 134 onde a economia de bits é comparada com zero. Se a economia de bits não for maior do que zero, as etapas de 108 a 118 são repetidas pelas etapas 136 a 146 com uma diferença. As etapas 110, 116 e 118 diminuem o cabeçalho porque para essas passagens o pixel sendo processado é um LPP, ao passo que nas etapas 138, 144 e 146 o pixel sendo processado não é um LPP e o cabeçalho não é diminuído.

Se a economia de bit for maior do que zero na etapa 134, as etapas 148 a 160 repetem as etapas de 120 a 132 que empregam a tabela de cinco bits e análise de preenchimento com zero. Novamente, uma vez que o pixel sendo processado pelas etapas 148 a 160 não é um LPP, o cabeçalho não é diminuído nas etapas 154, 156, 158 e 160. Após todas as etapas 138, 144, 146, 154, 156, 158 e 160, a compressão é completa e o próximo pixel é recuperado para processamento que começa em Início.

O objetivo do controlador 29 é processar cada pixel de croma com a tabela de quantificação de resolução o mais alto possível. Uma vez que os valores de dados de pixels são representações espaciais, cada valor de pixel contém informações importantes para a exibição, ao contrário de dados de blocos transformados que contêm a maior parte de suas informações no começo do bloco. É por isso que cada pixel é processado individualmente considerando-se sua posição relativa no bloco de pixels e o número de bits que o controlador 29 deve compensar.

Com referência de volta à Figura 1, a memória 14 armazena o bloco de pixels quantificado e o campo de parâmetros até que eles já não sejam necessários para a reconstrução de pixels e exibição. Durante o tempo em que os dados residem na memória 14, eles podem ser acessados e decodificados por um processador de exibição subsequente através do descompressor 16 sob o controle de um microprocessador que usa um barramento de dados comum. O compressor 12 e o descompressor 16 residem em um circuito integrado comum e exi-

bem um design e uma construção similares para simplificar o circuito integrado. A memória 14 fica vantajosamente situada do lado de fora do circuito integrado, de modo a permitir que o tamanho da memória 14 seja selecionado conforme necessário para acomodar os requisitos de processamento de sinal de um sistema particular. Isto resulta em economia de custo de fabricação, por exemplo, no caso de um receptor de televisão de consumidor de custo reduzido que usa uma exibição de resolução reduzida que exige menos memória de quadros para o decodificador MPEG. Além disso, muito embora a memória 14 possa ficar situada fora do circuito integrado, as arquiteturas de memória unificada do estado da técnica permitem que qualquer área de armazenamento não usada na memória 14 possa ser utilizada por outros componentes do sistema. Isto adiciona 15 redutor o custo do sistema total e aumenta a capacidade geral do sistema. Da mesma forma, a área de armazenamento economizada é tipicamente utilizável por outros componentes do sistema, o que aumenta a capacidade geral do sistema.

20 A Figura 10 exemplifica partes de um sistema de processamento de sinal digital prático em um receptor de televisão, que inclui um aparelho de acordo com a presente invenção conforme discutido previamente. O sistema receptor de televisão digital da Figura 10 é simplificado de modo a 25 não sobrecarregar o desenho com detalhe excessivo. Por exemplo, não se mostram a entrada FIFO e os buffers de saída associados com vários elementos, controles de leitura/gravação, redes de gerador de pulsos de relógio (clock) e

sinais de controle para formar interface com as memórias externas que podem ser do tipo saída de dados estendidos (EDO) ou tipo síncrono (SDRAM), DRAM Rambus (RDRAM) ou qualquer outro tipo de RAM.

5 Os elementos comuns na Figura 1 e Figura 10 têm o mesmo identificador. Os elementos no processador de sinal 72, exceto a unidade 70, correspondem aos elementos encontrados no circuito integrado Sti 3500A MPEG-2/CCIR 600 Decodificador de Vídeo comercialmente disponível junto à SGS-  
10 Thomson Microelectronics.

Resumidamente, o sistema da Figura 10 inclui o microprocessador 40, a unidade de interface de barramento 41 e o controlador 44 acoplado a um barramento de controle interno 46. Neste exemplo, o microprocessador 40 fica localizado 15 do lado externo do circuito integrado que contém o decodificador MPEG 72. Um barramento de memória interna com 192 bits de extensão 48 é um conduto para os dados do e para o compressor 12, os descompressores similares 16 e 50 e a memória de quadros externa 14. As unidades 12, 16 e 50 recebem 20 sinais de controle de fator de compressão e descompressão do microprocessador 40 pelo controlador 44, juntamente com os sinais de controle de ativação. Também é incluída uma unidade de controle de memória local 52 que recebe as entradas de Solicitação e envia as saídas de Reconhecimento 25 bem como as saídas de Endereço de memória, as saídas de Ativar Leitura e Ativar Gravação. A unidade de controle de memória 52 gera sinais de endereço em tempo real e de controle para controlar a memória 14. A unidade de controle de memó-

ria 52 também envia sinais de pulso horário de saída Clock Out em resposta aos sinais de pulso horário de entrada Clock In de um gerador de pulsos horários local (não mostrado). O microprocessador 40 partitiona a memória 14 em buffers de 5 bits, seções de armazenamento de quadro de vídeo e buffers de armazenamento de quadros para a decodificação MPEG, e mapas de processamento de exibição e exibição na tela.

O processador de exibição 54 inclui filtros de amostragem horizontal e vertical conforme necessário para 10 converter um formato de imagem descomprimido em um formato comum predeterminado para exibição por um dispositivo de exibição de reprodução de imagem 56. Por exemplo, o sistema pode receber e decodificar seqüências de imagem correspondentes a formatos tais como varredura entrelaçada de 525 linhas, varredura entrelaçada de 1125 linhas ou varredura progressiva de 720 linhas. Um receptor de televisão vai provavelmente usar um formato de exibição comum para todos os formatos do receptor.

As redes de interface externa 58 transportam informações de controle e configuração entre o decodificador MPEG e o microprocessador externo 40 além de inserir os dados de vídeo comprimidos para processamento pelo decodificador MPEG. O sistema decodificador MPEG se parece com um co-processador-processador para o microprocessador 40. Por exemplo, o microprocessador 40 edita um comando de decodificação para o decodificador MPEG para cada quadro a ser decodificado. O decodificador localiza as informações do cabeçalho associadas, as quais o microprocessador 40 então lê.

Com estas informações, o microprocessador 40 envia os dados para configurar o decodificador, por exemplo, com relação ao tipo de quadro, matrizes de quantificação, etc., após o que o decodificador envia os comandos de decodificação apropriados. 5 Os materiais de especificação técnica para o dispositivo de circuito integrado SGS-Thomson STi 3500A acima observado fornecem as informações adicionais com relação a essa maneira de operação do decodificador MPEG.

O microprocessador 40 transporta os dados de controle de modo, programado pelo fabricante do receptor, para o controlador de memória 52 para o controle da operação do multiplexador 32 (Figura 2) e o desmultiplexador 34 (Figura 4), e para estabelecer os fatores de compressão/descompressão para as unidades 12, 16 e 50, conforme necessário. 10 O sistema descrito pode ser usado com todos os Perfis e todos os Níveis da especificação MPEG no contexto de vários esquemas de processamento de dados digitais, tal como pode ser associado com sistemas de transmissão de difusão terrestre, via cabo e satélite, por exemplo. 15

A Figura 10 também retrata uma parte de um processador de sinal de vídeo digital 72, tal como pode ser encontrado em um receptor de televisão para processamento de um sinal de vídeo de alta definição de entrada. 20 O processador de sinal 72 pode ser incluído em um circuito integrado que inclui provisões para receber e processar sinais de vídeo de definição padrão através de um canal análogo (não mostrado). 25 O processador de sinal 72 inclui um decodificador MPEG convencional constituído pelos blocos 60, 62, 64, 66, 68 e 70,

inclusive memória de quadros 14. Por exemplo, Ang. et al. "Video Compression Makes Big Gains", IEEE Spectrum, outubro de 1991, descrevem a operação de um codificador e decodificador MPEG.

5 O processador de sinais 72 recebe uma corrente de dados controlados dos dados codificados em MPEG de um processador de entrada precedente (não mostrado), por exemplo, um decodificador de transporte que separa os pacotes de dados após a demodulação de sinal de entrada. Neste exemplo, 10 a corrente de dados de entrada recebida representa o material de imagem de alta definição (1920 x 1088 pixels) conforme especificado na especificação Grand Alliance para o sistema de difusão de televisão terrestre de alta definição dos Estados Unidos. A corrente de dados de entrada está na forma 15 de uma corrente de dados em camadas representando uma sequência de imagens que foi comprimida com o uso do padrão de compressão MPEG. Periodicamente, os blocos de dados representam informações comprimidas, codificadas intraquadros e interquadros. As informações intraquadros contém quadros de 20 âncora de quadro em "I". Geralmente, as informações interquadros contêm informações residuais codificadas de movimento previsíveis que representam a diferença de imagem entre os quadros de imagens adjacentes. A codificação do movimento interquadros envolve a geração de vetores de movimento 25 que representam o deslocamento entre um bloco atual sendo processado e um bloco em uma imagem previamente reconstruída. O vetor de movimento que representa a melhor correspondência entre os blocos atual e anterior é codificada e

transmitida. Da mesma forma, a diferença (residual) entre cada bloco de 8 x 8 de movimento compensado e o bloco previamente reconstruído é a transformada discreta de cosseno (DCT), quantificada e codificada por extensão variável (VLC) 5 antes de ser transmitida. Várias publicações, inclusive de Ang., et al. acima, descrevem os processos de codificação com movimento compensado com mais detalhes.

O buffer 60 aceita os blocos de dados de pixels comprimidos de entrada antes de ser decodificados com extensão variável pelo decodificador de extensão variável (VLD) 10 62. O buffer 60 exibe uma capacidade de armazenamento de 1,75 Mbit no caso de uma corrente de dados MPEG de perfil principal de nível principal. O quantificador inverso 64 e a transformada discreta inversa discreta de cosseno (IDCT) 15 66 descomprimem os dados comprimidos decodificados do VLD 62. Os dados de saída do IDCT 66 são acoplados a uma entrada do adicionador 68.

Um sinal do buffer 60 controla o tamanho da etapa de quantificação do quantificador inverso 64 para garantir 20 um fluxo de dados moderado. O VLD 62 fornece os vetores de movimento decodificado para a unidade de compensação de movimento 70, conforme será discutido. O VLD 62 também produz um sinal de controle de selecionar modo de inter/intra quadro conforme conhecido (não mostrado para simplificação). A 25 operação executada pelas unidades 62, 64 e 66 são operações correspondentes inversas de um codificador localizado em um transmissor.

Ao somar os dados de imagem residual da unidade 66 com os dados de imagem previstos obtidos a partir da saída da unidade 70, o adicionador 68 apresenta um pixel reconstruído com base nos conteúdos da memória de quadros de vídeo 5 14. Quando o processador de sinal 72 tiver processado um quadro inteiro de blocos de pixels, a memória de quadros 14 armazena a imagem reconstruída resultante. No modo inter-quadros, os vetores de movimento obtidos a partir do VLD 62 fornecem a localização dos blocos previstos da unidade 70.

10 O processo de reconstrução de imagem que envolve o adicionador 68, a memória 14 e a unidade de compensação de movimento 70 exibe vantajosamente requisitos de memória significativamente reduzidos devido ao uso do compressor de blocos 12 antes do armazenamento de dados na memória de quadros 14. O tamanho da memória de quadros 14 pode ser reduzido em até cinqüenta por cento (50%), por exemplo, quando um fator de compressão de 50% for usado. A unidade 50 desempenha a função inversa da unidade 12, e é similar ao descompressor 16 acima descrito. O descompressor 50 reconstrói 15 o bloco de imagem de modo que o compensador de movimento 70 possa funcionar conforme acima descrito. O compressor 12 e os descompressores 16 e 50 são construídos de acordo com os princípios da presente invenção. As Figuras 1, 2, 4, 5A e 7 20 ilustram detalhes dentro das unidades 12, 16 e 50.

REIVINDICAÇÕES

1. Aparelho em um sistema de processamento de  
imagem digital para receber uma corrente de dados de blocos  
de pixels de imagem comprimidos, **CARACTERIZADO** pelo fato de  
5 compreender:

um dispositivo (72) para descomprimir os referidos  
blocos de pixels comprimidos;

10 um circuito (20) para recomprimir representações  
de pixels de um bloco de pixels descomprimidos de modo que o  
número médio de bits para uma representação de pixels de  
luminância comprimida seja pelo menos um bit maior do que  
10 para uma representação de pixels de crominância comprimida;

uma memória (14) para armazenar as referidas  
representações de pixels recomprimidas;

15 um dispositivo de previsão para produzir valores  
de erro de previsão a partir das referidas representações de  
pixel;

um segundo circuito para derivar um parâmetro de  
bloco do referido bloco de pixel;

20 um processador de previsão de erro para polarizar  
valores negativos de previsão de erro para produzir valores  
positivos antes da recompressão; e

25 um controlador para garantir um bloco de pixels  
recomprimidos e o referido parâmetro de bloco igual a um  
número predeterminado de bits.

2. Aparelho, de acordo com a reivindicação 1,  
**CARACTERIZADO** pelo fato de que:

o segundo circuito é adaptável à faixa de valores  
de pixel do referido bloco de pixel para estabelecer faixas  
5 e selecionar uma faixa predeterminada; e

a faixa predeterminada acessa um conjunto de  
tabelas de compressão customizadas para a referida faixa  
predeterminada.

3. Aparelho, de acordo com a reivindicação 2,  
10 **CARACTERIZADO** pelo fato de que as referidas tabelas  
customizadas incluem palavras de código curtas.

4. Aparelho, de acordo com a reivindicação 2,  
**CARACTERIZADO** pelo fato de que as referidas tabelas  
customizadas são simétricas.

15 5. Aparelho, de acordo com a reivindicação 1,  
**CARACTERIZADO** pelo fato de que o referido parâmetro de bloco  
está codificado e arranjado como um campo de parâmetros do  
referido bloco de pixels recomprimidos e que o campo de  
parâmetros e o bloco de pixels comprimidos referidos formam  
20 um pacote de dados.

6. Aparelho, de acordo com a reivindicação 1,  
**CARACTERIZADO** pelo fato de que o referido parâmetro de bloco  
do referido bloco de pixel é um dentre o valor de pixel  
mínimo, do valor de pixel máximo, do referido valor de pixel

de referência, a referida faixa e a referida faixa predeterminada.

7. Aparelho em um sistema de processamento de imagem digital para processar blocos de pixels de imagem codificada por MPEG que contém dados de luminância e crominância, sendo que os dados de luminância são representados por mais pixels do que os ditos dados de crominância, **CARACTERIZADO** pelo fato de compreender:

um dispositivo descompressor (72) para 10 descomprimir blocos de pixels comprimidos;

um circuito (20) para recomprimir representações de pixels de um bloco de pixels descomprimidos para representações de pixel recomprimidas; e uma memória de quadros (14) para armazenar as representações de pixel 15 recomprimidas sendo que o dito bloco de pixels descomprimido contém dados de luminância e crominância recomprimidos a um número médio predeterminado de bits, onde o número médio de bits para uma representação de pixels de luminância é pelo menos um bit maior do que para uma representação de pixels 20 de crominância;

uma rede de previsão para processar os aslores de erro de previsão;

um descompressor para descomprimir as ditas representações de pixels recomprimidas para pixels 25 reconstruídos; e

um processador de exibição para receber as informações de pixels da dita memória a um dispositivo de exibição de imagem que exibe uma resolução maior do que a resolução NTSC padrão, sendo que as representações de pixels recomprimidas são os referidos valores de erro de previsão;

os referidos valores de erro de previsão são valores diferenciais de modulação por pulsos codificados; e a referida recompressão é facilitada pela polarização de erros negativos de previsão para erros positivos de previsão.

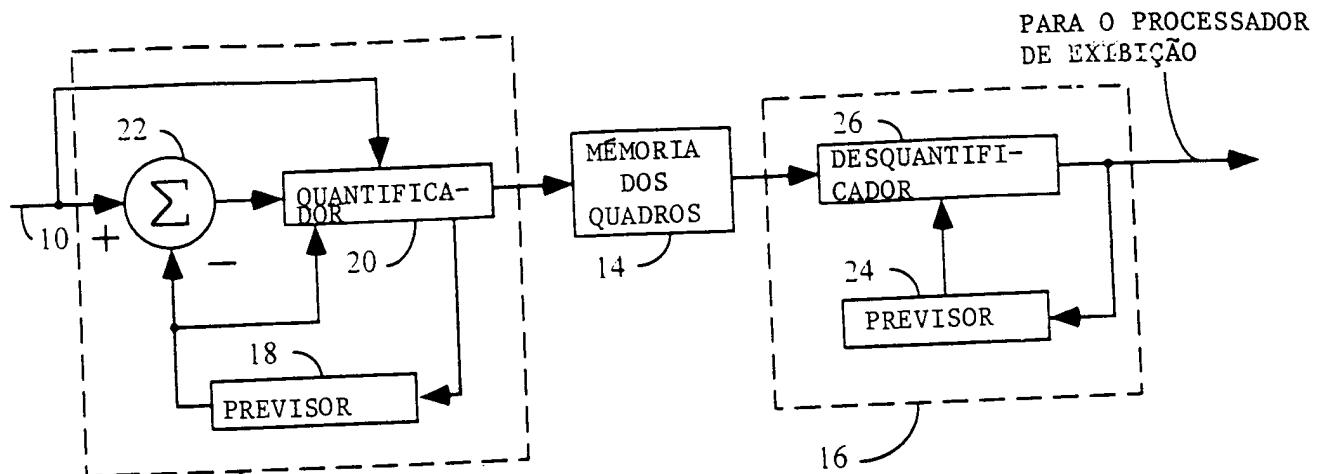


FIG. 1

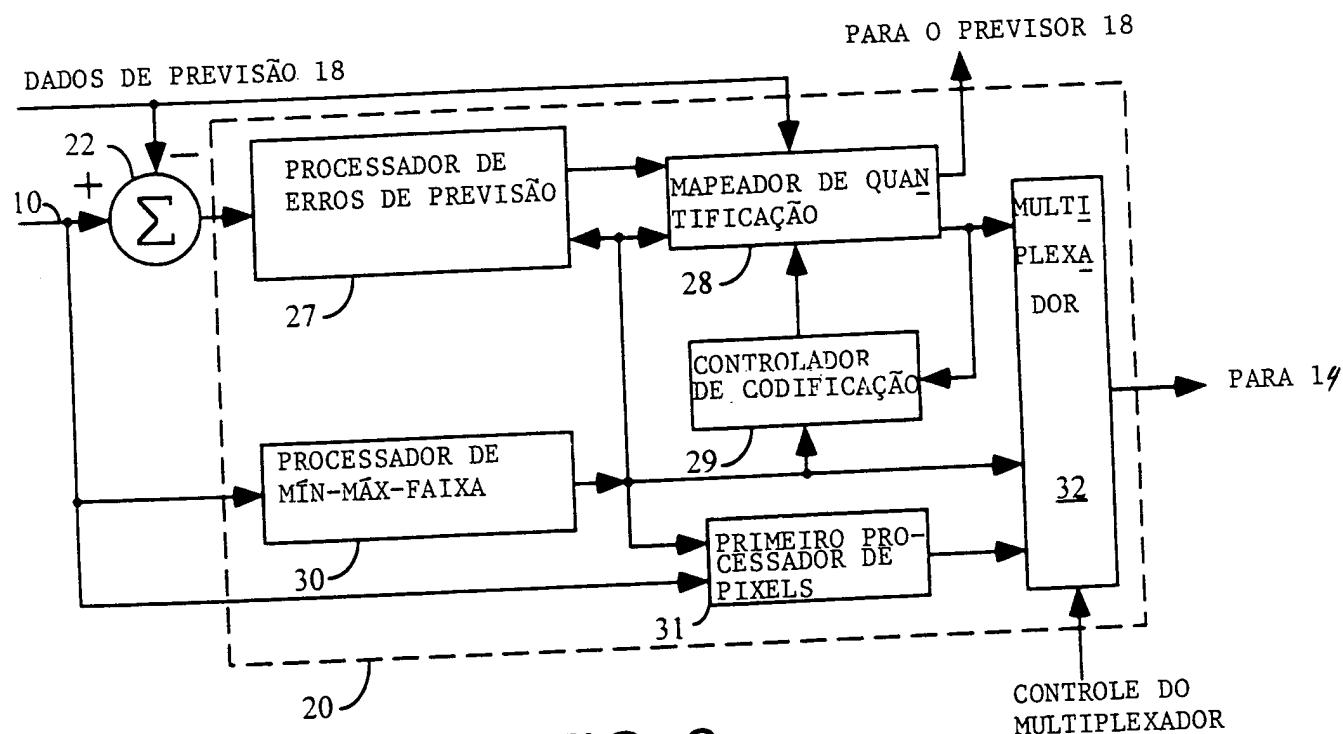


FIG. 2

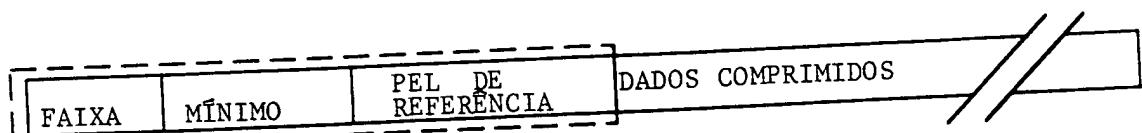


FIG. 3

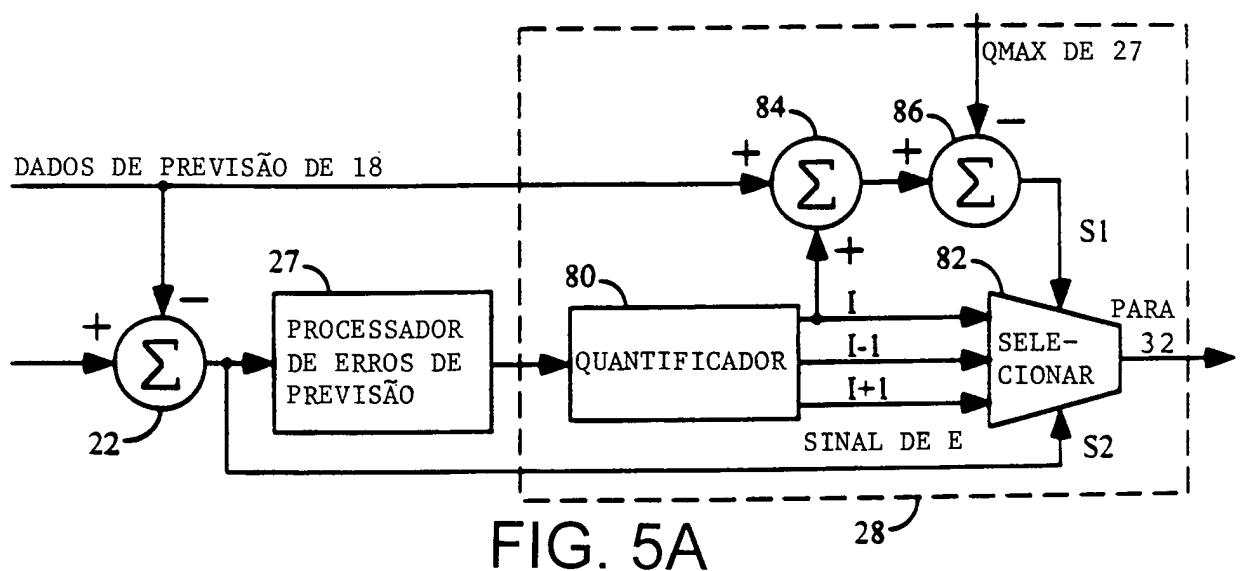
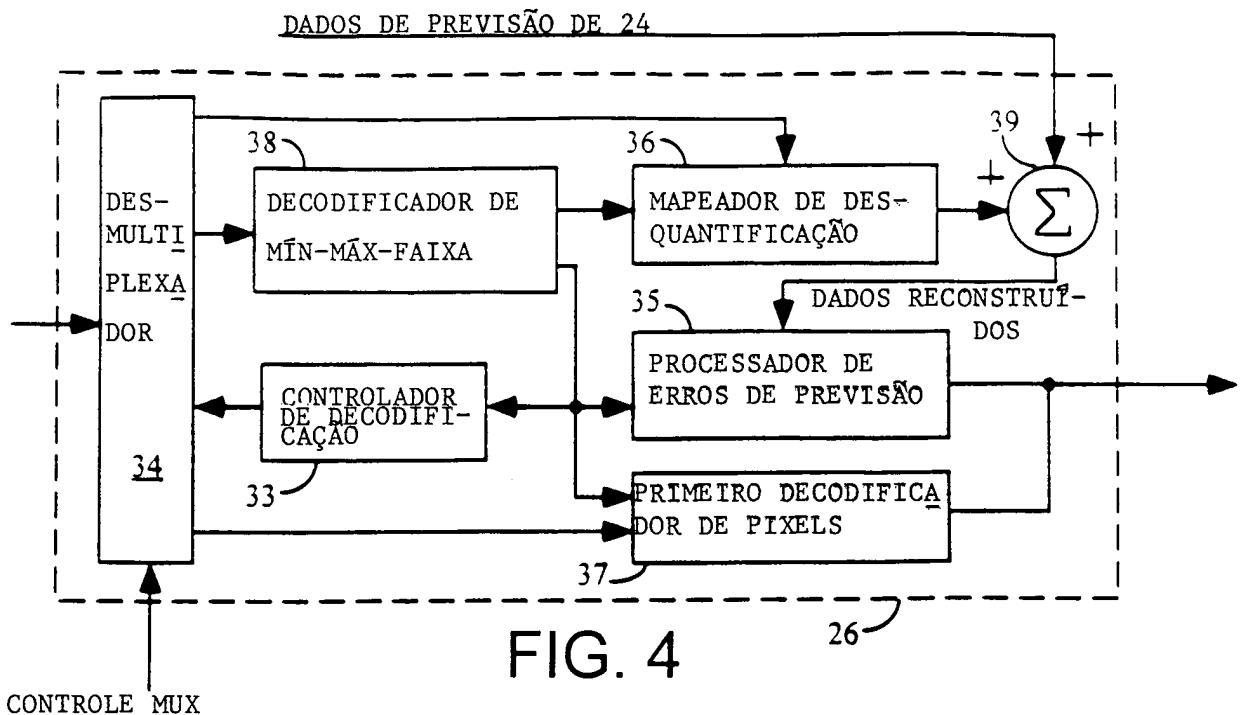


FIG. 5B

	+ S1	- S1
+ S2	I-1	I
- S2	I	I+1

ÍNDICE	PONTO DE DECISÃO	NÍVEL DE RECON	PALAVRA DE CÓDIGO
0	1	0	1110
1	4	3	000
2	7	6	0100
3	10	9	0110
4	13	11	1000
5	19	16	1010
6	27	23	1100
7	36	32	1111
8	44	41	1101
9	50	48	1011
10	53	53	1001
11	56	55	0111
12	59	58	0101
13	62	61	001

FIG. 6B

ÍNDICE	PONTO DE DECISÃO	NÍVEL DE RECON	PALAVRA DE CÓDIGO
0	1	0	11110
1	3	2	000
2	5	4	01000
3	7	6	01010
4	9	8	01100
5	11	10	01110
6	13	12	10000
7	15	14	10010
8	17	16	10100
9	19	18	10110
10	21	20	11000
11	23	22	11010
12	29	26	11100
13	34	32	11111
14	40	38	11101
15	42	42	11011
16	44	44	11001
17	46	46	10111
18	48	48	10101
19	50	50	10011
20	52	52	10001
21	54	54	01111
22	56	56	01101
23	58	58	01011
24	60	60	01001
25	62	62	001

FIG. 6C

ÍNDICE	PONTO DE DECISÃO	NÍVEL DE RECON	PALAVRA DE CÓDIGO
0	2	0	110
1	8	5	000
2	16	12	010
3	25	20	100
4	38	32	111
5	47	44	101
6	55	52	011
7	61	59	001

FIG. 6A

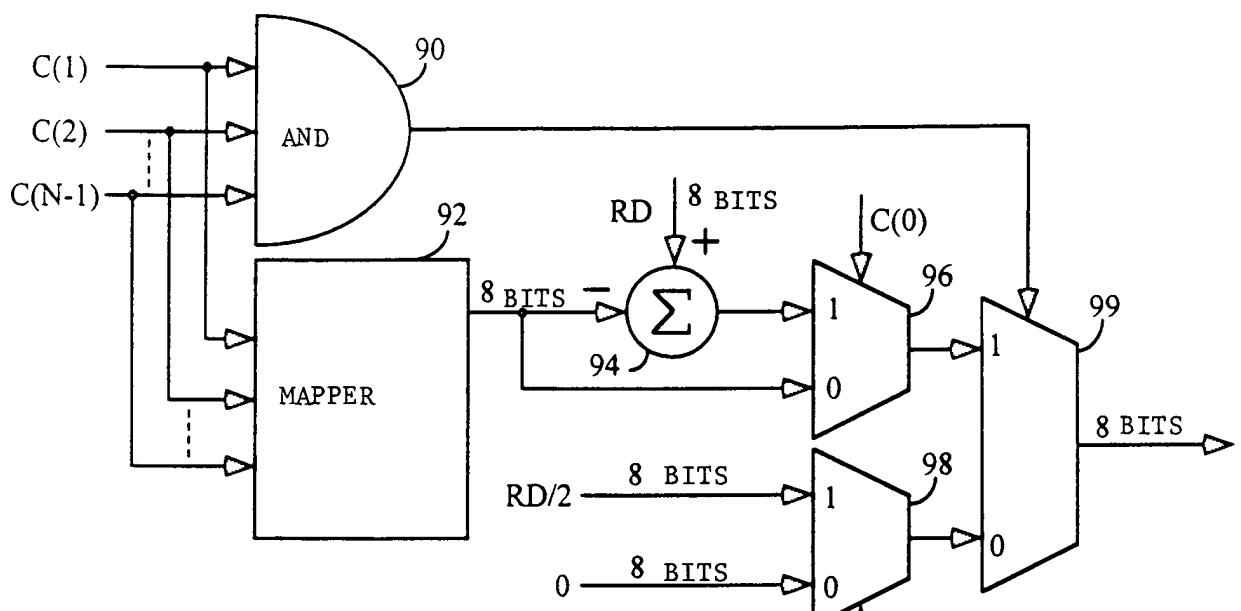
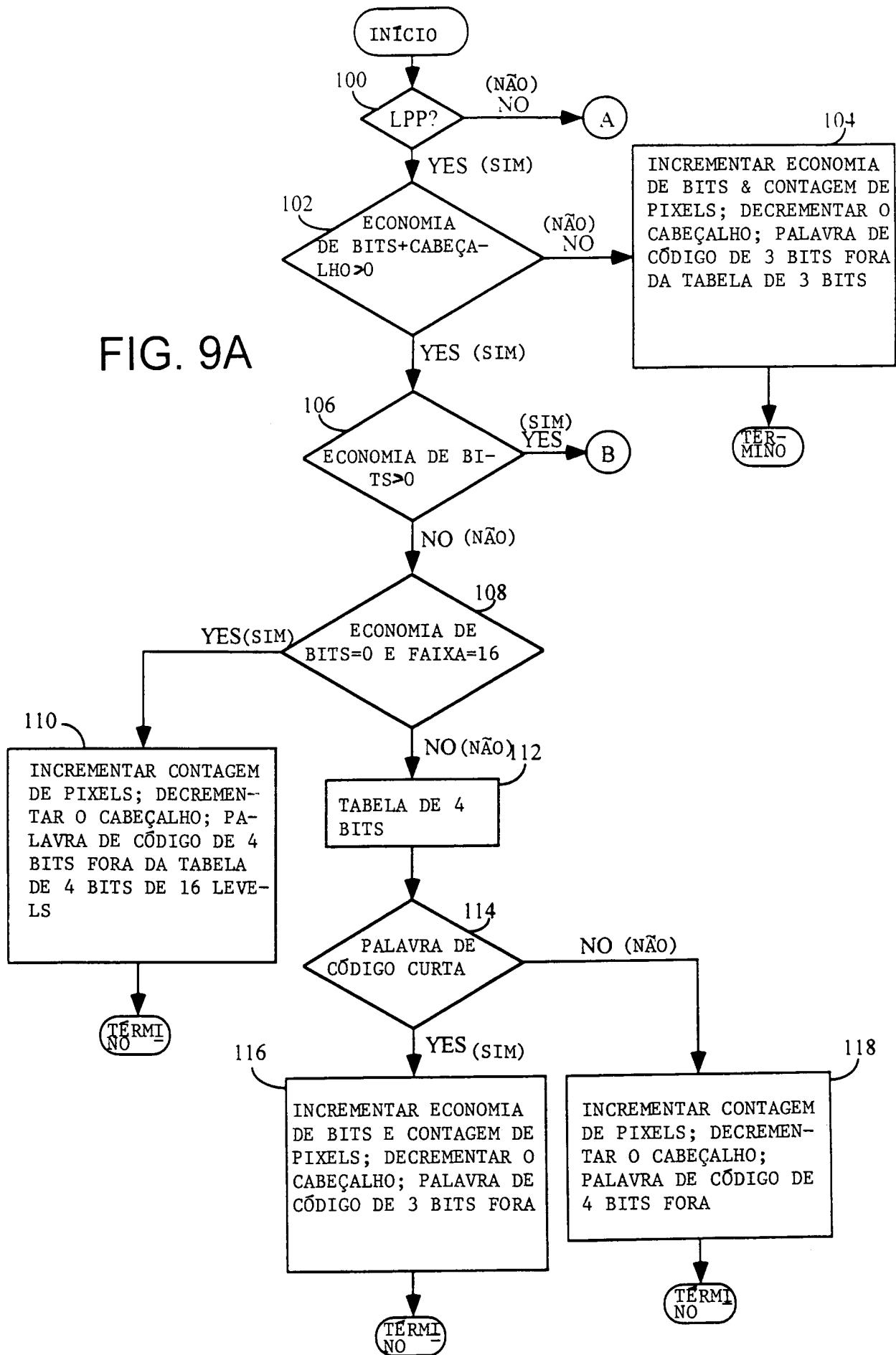
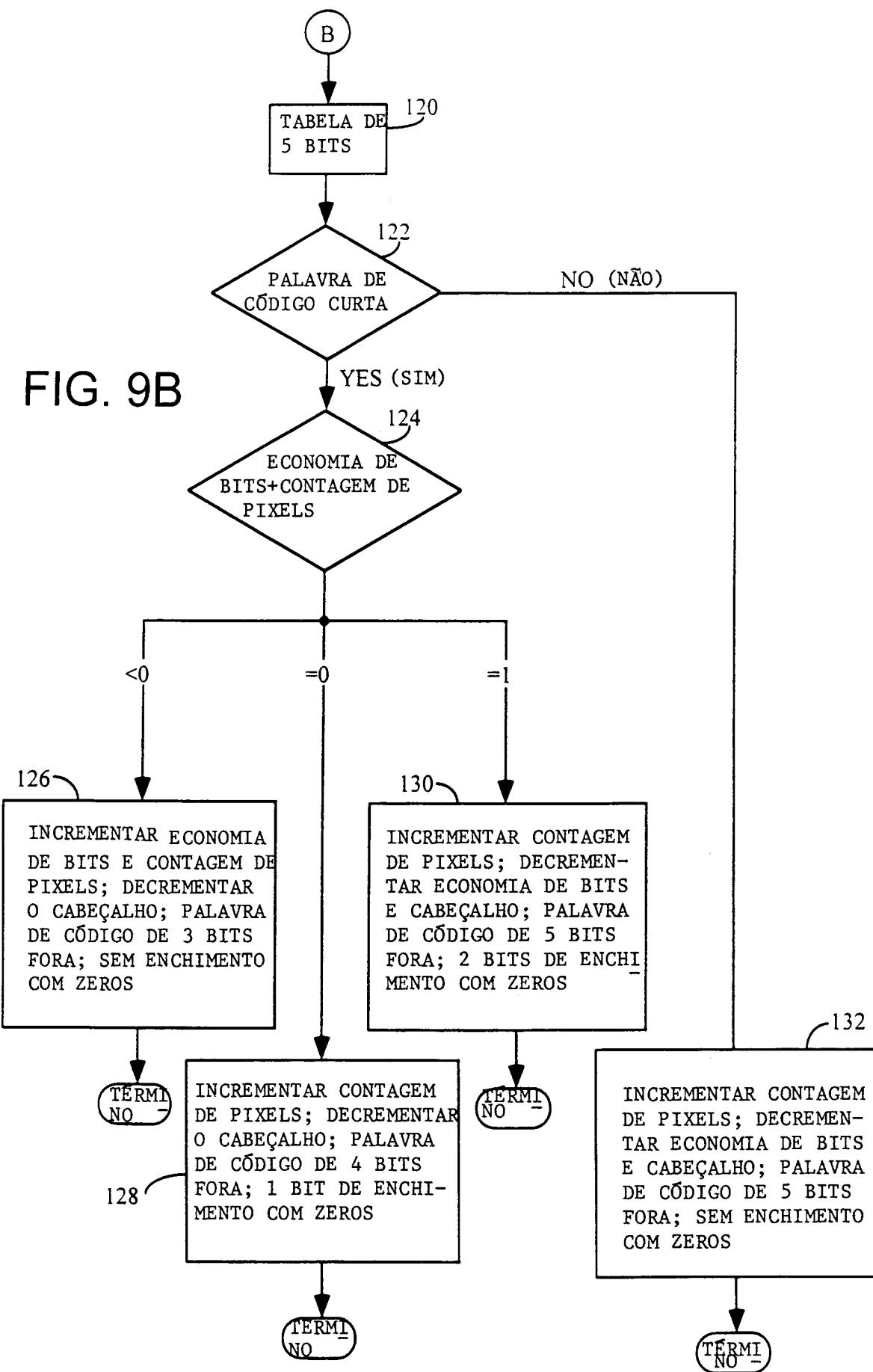


FIG. 7

FAIXA PREDETERMINADA	16	32	64	96	128	196	256
FAIXA DE BITS	3	3	3	3	3	3	3
MÍNIMO DE BITS	8	3	3	3	3	3	0
BITS DE PIXELS DE REFERÊNCIA	3	4	5	6	6	7	7
BITS SUSPENSOS	10	6	7	8	8	9	6

FIG. 8





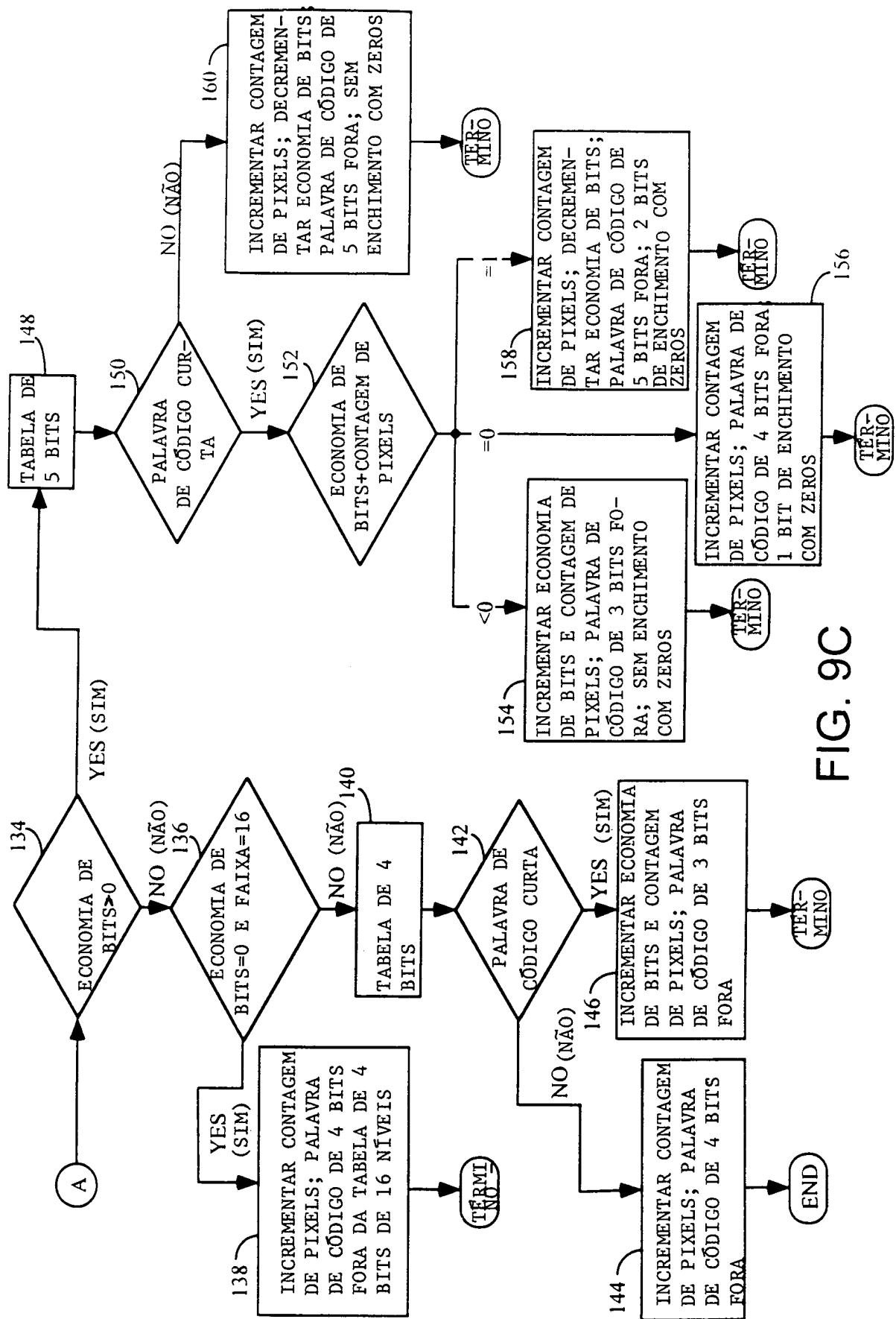


FIG. 9C

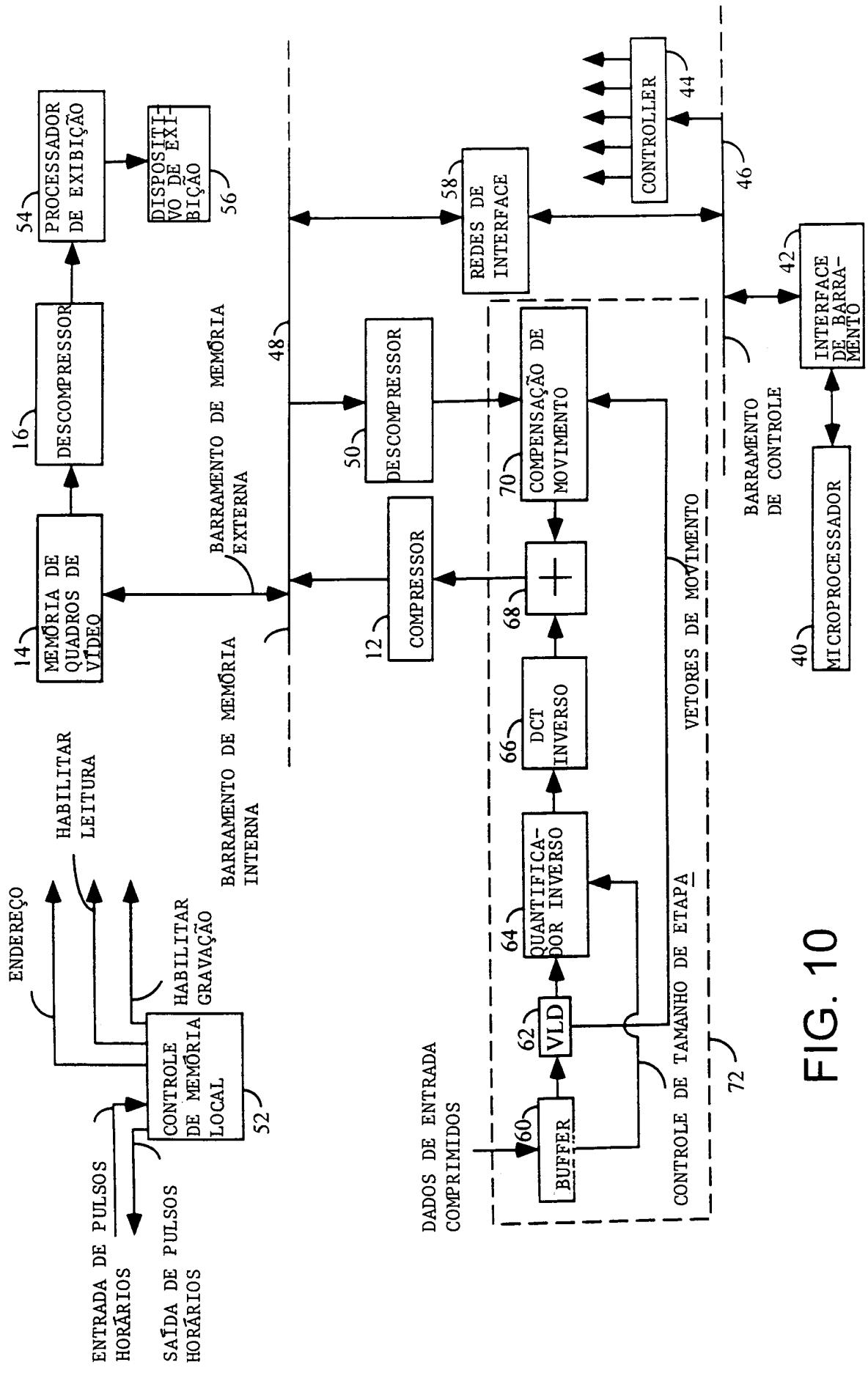


FIG. 10

## RESUMO

### "APARELHO DE COMPRESSÃO DE BLOCOS DE PIXELS EM UM SISTEMA DE PROCESSAMENTO DE IMAGEM"

Um processador de imagem com economia de memória 5 (20) recebe valores de erro de previsão de DPCM de sinais de vídeo digital codificado em MPEG descomprimido na forma de blocos de pixels que contêm dados de luminância e crominância em um formato 4:2:2 ou 4:2:0 e recomprime os blocos de pixels a uma resolução predeterminada. Os dados de luminância e crominância são processados com leis de compressão diferentes durante a recompressão. Os dados de luminância são recomprimidos a uma média de seis bits por pixel, e somente 10 um pixel de referência e um outro pixel são processados separadamente a partir de todos os outros pixels de luminância 15 em um bloco. Os dados de crominância são recomprimidos a uma média de quatro bits por pixel. Cada bloco de pixels é armazenado com informações suspensas que facilitam a reconstrução eficiente e precisa. A reconstrução precisa de pixels é facilitada pelo processamento de um pixel de referência com precisão (31); graduação do blocos de pixels (28); 20 emprego de tabelas de quantificação (28) que são simétricas e ajustadas ao domínio do blocos de pixels; compensação de valores de erro de previsão negativos (28) em valores positivos; uso de palavras de código em tabelas de quantificação 25 (28) a níveis mais prováveis de ocorrer estatisticamente; e processamento de cada pixel com quantificação de três, quatro ou cinco bits (28) para garantir a resolução máxima e uma média geral de quatro bits para o blocos de pixels.