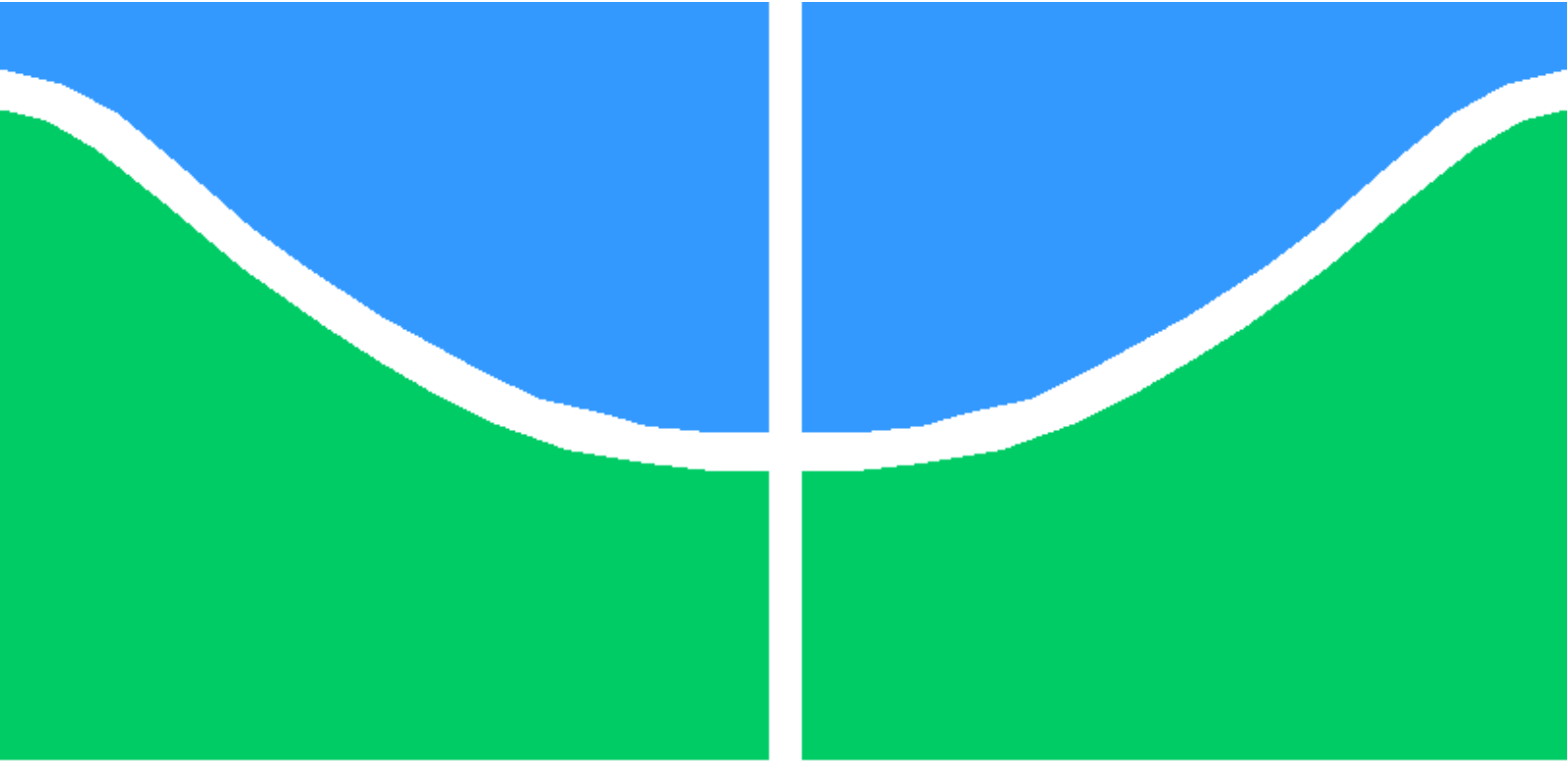


TESE DE DOUTORADO EM ENGENHARIA DE SISTEMAS ELETRÔNICOS E AUTOMAÇÃO

REALCE DE VÍDEO PARA SEQÜÊNCIAS DE QUALIDADE E RESOLUÇÃO VARIÁVEIS

Edson Mintsu Hung

Brasília, agosto de 2012

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

REALCE DE VÍDEO PARA SEQÜÊNCIAS DE QUALIDADE E RESOLUÇÃO VARIÁVEIS

Edson Mintsu Hung

ORIENTADOR: Ricardo Lopes de Queiroz

TESE DE DOUTORADO EM ENGENHARIA DE SISTEMAS ELETRÔNICOS E AUTOMAÇÃO

Publicação: PPGEA.TD 060/2012
Brasília/DF: Agosto-2012

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

TESE DE DOUTORADO EM ENGENHARIA DE SISTEMAS ELETRÔNICOS E AUTOMAÇÃO

REALCE DE VÍDEO PARA SEQÜÊNCIAS DE QUALIDADE E RESOLUÇÃO VARIÁVEIS

Edson Mintsu Hung

Tese de doutorado submetida ao Departamento de Engenharia Elétrica da Faculdade de Tecnologia da Universidade de Brasília, como parte dos requisitos necessários para a obtenção do grau de doutor.

Banca Examinadora

Prof. Ricardo Lopes de Queiroz, PhD. _____
UnB/ CIC (Orientador)

Prof. Eduardo A. Barros da Silva, PhD. _____
UFRJ/ COPPE (Examinador Externo)

Prof. Alexandre Zaghetto, Dr. _____
UnB/ CIC (Examinador Externo)

Prof. Francisco Assis de O. Nascimento, Dr. _____
UnB/ ENE (Examinador Interno)

Prof. João Luiz Azevedo de Carvalho, PhD. _____
UnB/ ENE (Examinador Interno)

Prof. Bruno Luiggi Macchiavello Espinoza, Dr. _____
UnB/ CIC (Suplente)

FICHA CATALOGRÁFICA

HUNG, EDSON MINTSU
Realce de Vídeo para Seqüências de
Qualidade e Resolução Variáveis. [Distrito Federal] 2012.
xxii, 118p., 297 mm (ENE/FT/UnB, Doutorado, Telecomunicações
Processamento de Sinais, 2012). Tese de Doutorado.
Universidade de Brasília. Faculdade de Tecnologia.
Departamento de Engenharia Elétrica.

1. Super-resolução baseada em exemplos	2. Compressão de vídeo
3. Estimação de movimento	4. Blocos sobrepostos
5. Transformada de Cossenos Discreta	6. Quantização
I. ENE/FT/UnB	II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

HUNG, E. M. (2012). Realce de Vídeo para Seqüências de Qualidade e Resolução Variáveis. Tese de Doutorado em Engenharia de Sistemas Eletrônicos e Automação, Publicação PPGEA.TD - 060/2012, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 118p.

CESSÃO DE DIREITOS

NOME DO AUTOR: Edson Mintsu Hung.

TÍTULO DA TESE DE DOUTORADO: Realce de Vídeo para Seqüências de Qualidade e Resolução Variáveis.

GRAU / ANO: Doutor / 2012

Edson Mintsu Hung
QNL 11 Blobo E casa 06
72.151-115 Taguatinga - DF - Brasil.

Dedicatória

À minha família, meu pai Hung Chao-Shiung, à memória de minha mãe Yang Yu-Chu Hung, meus irmãos Alexandre Hung e Augusto Hung, minha sobrinha Júlia Yang de Oliveira Hung e à memória de meu sobrinho João Gabriel de Oliveira Hung, por participarem da minha vida. Aos meus amigos, pela compreensão.

Edson Mintsu Hung

“Se você tem uma laranja e troca com outra pessoa que também tem uma laranja, cada um fica com uma laranja. Mas se você tem uma idéia e troca com outra pessoa que também tem uma idéia, cada um fica com duas.”

Confúcio

Agradecimentos

*Agradeço ao povo brasileiro por financiar minha formação na Universidade de Brasília (**UnB**). Gostaria também de agradecer a UnB, ao Departamento de Engenharia Elétrica (**ENE**) e ao SG-11, pela qualidade dos **docentes** e **funcionários** que me acompanharam diariamente. Em destaque os professores que me incentivaram Francisco Assis Oliveira Nascimento, Geovany Araújo Borges, Ricardo Zelenovsky, Anderson Nascimento, Adson Ferreira da Rocha, Pedro Berger e João Ishihara.*

*Um agradecimento especial ao meu professor **orientador** Ricardo Lopes de Queiroz, pessoa pelo qual tenho muita admiração. Sua "paciência", dedicação e competência foram fundamentais para que esta tese se concretizasse.*

*Aos integrantes do Grupo de Processamento Digital de Sinais - **GPDS**, pelo apoio, amizade e companheirismo. Principalmente aos amigos: Alberto L. Delis, Alexandre Zaghetto, Bruno Machiavello, Camilo Chang Dórea, Eduardo Peixoto, Fabiano Soares, Fernanda Brandi, Jorge Cormane, Karen França, Marcelo Villegas, Marcus Chaffim, Rafael Galvão, Rafael Ortis, Renan Utida, Thacio Scandaroli e Tiago Alves. Agradeço a todos pelos momentos de distração, amizade, companhia, conversas, discussões, reflexões, piadas, estórias, "causos" e mais um monte de coisas.*

*Muito obrigado ao Ricardo L. de Queiroz, Renan Utida e Camilo C. Dórea por revisarem as diversas versões desta tese e aos **membros da banca** por gentilmente aceitarem participar da defesa.*

Agradecimentos especiais aos amigos Eduardo Peixoto, Carol Viana, Fernanda Brandi, Maíra Lioi, Rafael Galvão e Érica Zaiden pelos passeios e viagens que fizemos. Obrigado também pela hospedagem. :-)

Agradeço ao doutor Debargha Mukherjee ex-HP Labs - Palo Alto, agora Google Inc. *por apoiar este trabalho. Agradeço a **HP Brasil**, **Finatec** e **FAP-DF** pelo suporte financeiro para a realização e divulgação de trabalhos relacionados à esta tese.*

*Aos fundadores da **Mux Engenharia**: Wagner Popov, Tiago Alves e Luis Prata, que sempre contaram com a minha ausência - os meus sinceros muito obrigado pela compreensão e pela paciência.*

*Aos meus **amigos** de longa data um forte abraço. Peço ainda desculpa pela(s) ausência(s)... aos amigos que fiz ao longo da jornada aqui na UnB. Que mesmo não fazendo parte deste universo acadêmico, vocês foram fundamentais para garantir minha lucidez.*

Ao amigos e colegas que fiz no campus da UnB no Gama.

*Ao pessoal do **Só Alegria Moto Clube** pela receptividade e amizade, mesmo eu não tendo moto (...ainda).*

*Por fim, agradeço a receptividade e amizade das famílias Cunha, Silva, Rangel, Santos, Oliveira, Carpaneda, Reis e Brandi. E à minha **família** Hung, que me apoiou durante toda vida. Agradeço por tudo, tanto pelos momentos bons, ruins e os mais difíceis.*

Edson Mintsu Hung

RESUMO

As técnicas propostas nesta tese permitem realçar a qualidade (objetiva e subjetiva) na decodificação de vídeo. Estas técnicas baseiam-se no uso de exemplos, também denominado quadros-chave que são imagens ou quadros com qualidade ou resolução que sejam maiores que a do vídeo alvo, denominados de quadros-não-chave. Neste caso, serão compostos dicionários contendo informações (exemplos) dos quadros-chave para super-resolver, ou realçar, os quadros do vídeo com baixa-resolução ou qualidade, denominados de não-chaves. As arquiteturas de qualidade e resolução mista podem ser adotadas em vários cenários como: a redução da complexidade durante o processo de compressão, redução da taxa de transmissão, melhorias na qualidade geral do vídeo baseadas em outros quadros ou imagens, correção de erros de transmissão, etc. Nesta tese são propostas duas novas técnicas utilizadas no processo de super-resolução baseada em exemplos: compensação de movimento utilizando blocos multi-escala sobrepostos e a combinação das informações de múltiplos dicionários. Um novo processo de extração de informação para aplicação em super-resolução utilizando o domínio transformado (DCT) também é proposto na tese. Por fim, propõe-se uma generalização do processo de realce baseado em exemplos para aplicação em vídeos com variação de qualidade entre quadros. Dentre as possíveis variações de qualidade foram contemplados: parâmetros de quantização (definindo a qualidade da compressão), foco ou ruído. Os cenários de aplicação testados nesta tese são: (i) vídeo com resolução mista, (ii) vídeos com múltiplas vistas em resolução mista com informação de profundidade, (iii) vídeo com fotografias redundantes durante a gravação, (iv) vídeo com qualidade mista.

ABSTRACT

This thesis proposes techniques for example-based enhancement of decoded video, providing both subjective and objective increases in quality. The techniques rely on the usage of information from images or frames available at greater quality or resolution (key-frames) to enhance the target images of lower quality or resolution (non-key-frames) within a video sequence. A codebook is composed of examples taken from the key-frames. From these examples, high-frequency information is extracted in order to enhance or super-resolve non-key-frames within video. This mixed quality or mixed resolution architecture may be adopted for appliations such as encoding complexity reduction, transmission bit-rate reduction, video enhancement based on other frames or images, error concealment, etc. In this thesis we first propose two techniques for usage in example-based super-resolution: a multi-scale overlapped block motion compensation scheme and a codebook combination of multiple dictionaries. Next, a novel transform-domain super-resolution method using the DCT is presented. Finally, a generalization of the example-based enhancement method is proposed. The generalization can account for videos with varying quality among frames due to different quantization parameters (which define the compression quality), focus or noise. The application scenarios considered in this thesis are: (i) mixed resolution video, (ii) multiview video plus depth with mixed resolution, (iii) videos with redundant snapshots, and (iv) mixed quality video.

SUMÁRIO

LISTA DE FIGURAS

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

A compressão de vídeo digital é uma codificação de fonte onde são almejadas boa qualidade das imagens e diminuição da quantidade de bits armazenados ou transmitidos. [3–5]

As cenas capturadas envolvem amostrar o espaço, projetando-o em um plano bidimensional, do ponto de vista da câmera, formando assim uma imagem ou quadro. Este processo é repetido em diversos instantes de tempo de forma a obter uma amostragem temporal da cena. A Figura 1.1 ilustra um vídeo amostrado, composto pelos menores elementos de uma imagem, denominados pixels (*picture elements*). No padrão YUV, os pixels possuem informações de brilho e cor, denominados respectivamente de luminância (Y) e crominância (U e V). A luminância gera uma seqüência de vídeo em escala de cinza, também conhecida como monocromática. Já as informações de crominância associam à luminância a saturação e a matiz das cores.

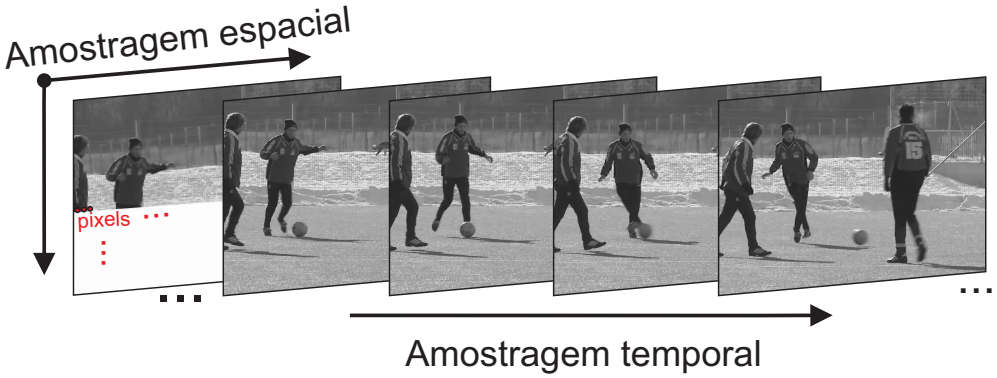


Figura 1.1: Seqüência de vídeo cuja cena capturada é amostrada.

Entretanto novas tecnologias de vídeo 3D baseadas em sistemas estéreo tem surgido no mercado. Estes sistemas de vídeo estéreo utilizam quadros codificados originados de duas câmeras. Tipicamente, o receptor reproduz apenas duas vistas sendo necessários *displays* polarizados e óculos 3D para que cada vista mantenha uma correspondência para cada olho do observador. Entretanto, existe uma outra tecnologia que dispensa o uso de óculos 3D, denominada autoestereoscópica. Em alguns casos, para que esse sistema seja viável, um número maior de vistas deve ser utilizado, sendo necessária uma extensão da codificação de vídeo estéreo. Contudo, para representar um número maior de vistas, temos um aumento de taxa de

bits proporcional ao número de vistas representadas. Portanto, um outro formato onde o sinal de vídeo é combinado com o mapa de profundidade foi proposto pelos pesquisadores [6, 7]. Onde, além do vídeo, é feito o uso de uma informação adicional que relaciona a distância entre a câmera e os objetos em cena. Os mapas de profundidade informam as disparidades associadas a cada amostra do sinal de vídeo e podem ser utilizados para renderizar um número arbitrário de novos pontos de vistas adicionais por meio da síntese de vistas entre os pontos de vistas capturados. Conforme ilustra a Figura 1.2, a captura é similar a um conjunto de vídeos com uma única vista. Os mapas de profundidade podem ser estimados utilizando algoritmos que calculam o posicionamento dos objetos em cena a partir de alguns conhecimentos *a priori*, como por exemplo a distância entre as câmeras, os parâmetros de calibração, etc. Uma outra possibilidade de se obter tal informação seria por meio de sensoriamento, que pode ser feito com o infra-vermelho no caso do Kinect [1] , por exemplo. Grande parte desses sensores funcionam como os sonares, onde uma onda é emitida e o tempo de reflexão desta onda é calculado, possibilitando assim o cálculo da distância entre o sensor e os objetos. A seqüência de vídeo com múltiplas vistas ainda é alvo de desenvolvimento na indústria, pois permite uma interação diferenciada, oferecendo a sensação de observar objetos tridimensionais, ou assistir um vídeo em pontos de vistas sintéticos (isto é, observar por um ponto diferente dos filmados), etc.

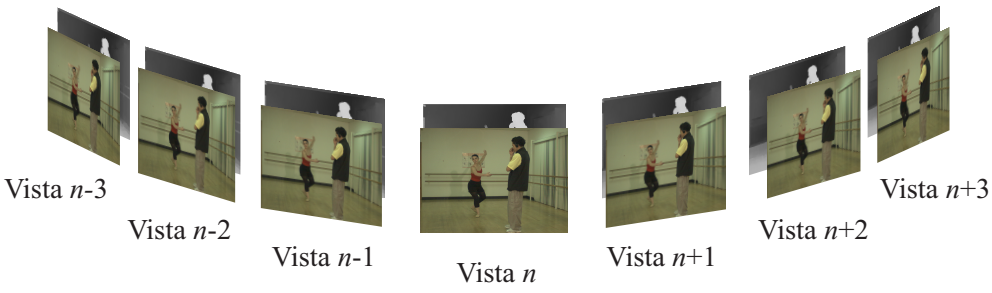


Figura 1.2: Seqüência de vídeo com múltiplas vistas e mapas de profundidade.

Observe, na Figura 1.1, que a redundância espacial e temporal em um vídeo é muito grande devido à correlação espacial que advém da proximidade de pixels vizinhos e ao grande conjunto de imagens capturadas em um pequeno intervalo de tempo, possuindo, no geral, diferenças relativamente pequenas entre si. Essas características são exploradas para que o vídeo digital seja representado e comprimido de maneira eficiente. Tipicamente, utilizam-se as técnicas de compressão de imagens para reduzir a redundância espacial, conhecidas na área de compressão de vídeo como codificação *intra*-quadros.

[1] O Kinect é um produto desenvolvido pela *Microsoft* para seu console de videogame Xbox 360 que permite os jogadores uma interação com os jogos sem a necessidade de controle ou *joystick*. O dispositivo possui uma câmera RGB, sensor de profundidade, um *array* de microfones e diversos softwares proprietários.

Entretanto, a redundância temporal é geralmente explorada por meio de estimação e compensação de movimento, gerando a codificação *inter*-quadros.

No caso de seqüências de vídeo com múltiplas vistas, como indica a Figura 1.2, a correlação entre as vistas também pode explorada ao fazer um re-ordenamento temporal nos quadros em uma única vista. Esse processo é denominado de *multicast* para o caso de transmissão (*streaming*). Caso se opte apenas por explorar as correlações espaço-temporais, o tipo de transmissão passa a ser *simulcast* [6]. No entanto, não existe uma padronização que defina o método mais eficaz de compressão de vídeo 3D com mapa de profundidade. Entretanto, exitem alguns estudos recentes [8–10] que tratam de estudar este tópico.

Atualmente, grande parte dos padrões de compressão de vídeo foram projetados para um sistema de radiodifusão (*broadcasting*), onde um codificador com grande capacidade de processamento comprime e envia a informação para vários decodificadores com baixa capacidade computacional. Outros padrões são mais específicos para codificação de vídeo com baixas taxa e latência, aplicados à videoconferência [11, 12]. Os padrões de codificação de vídeo convencionais, como o H.264/AVC [5], exigem uma grande complexidade computacional (capacidade de processamento e memória) no codificador se compararmos com o processo de decodificação [11]. Isto se deve às operações de predições *intra* e *inter* (estimação de movimento). Para que o H.264 atinja um bom desempenho de compressão, estes modos de predição são exaustivamente testados e escolhidos a partir da minimização de uma função de custo que relacionam a taxa de bits e a distorção do vídeo. No decodificador as predições e outras informações são apenas interpretadas, de forma a reconstruir de maneira aproximada (nos casos de codificação com perdas) o mesmo vídeo que foi codificado.

Outros requisitos também relacionados a aplicações em codificação de vídeo digital têm sido alvo de pesquisa. Requisitos como flutuação na taxa de transmissão, restrições de tempo (latência), complexidade do algoritmo, consumo de energia, qualidade de serviço (QoS) e robustez a erros, podem ser considerados tão importantes quanto a qualidade do vídeo e a taxa de compressão. Para tanto alguns cenários onde vídeos com resolução mista [11, 13] ou vídeos com qualidade mista [14] são estudados na literatura, de forma a permitir a escalabilidade da complexidade ou a flutuação da taxa de transmissão.

Na Seção 1.2 são descritos alguns casos onde vídeos com resolução e qualidade mista são utilizados. Estes vídeos são ilustrados nas Figuras 1.3 e 1.4, onde possuem variação espacial no tempo e no espaço, respectivamente. Já a Figura 1.5 ilustra um vídeo com qualidade temporalmente mista. Nesta tese, propõe-se utilizar o realce de vídeo, que é uma técnica de pós-processamento utilizado para acentuar algumas

características relevantes. Neste caso, o objetivo é aumentar a qualidade percebida de um vídeo utilizando as informações do quadro com resolução ou qualidade maior para realçar os quadros com resolução ou qualidade menor.

Figura 1.3: Seqüência de vídeo com resolução mista.

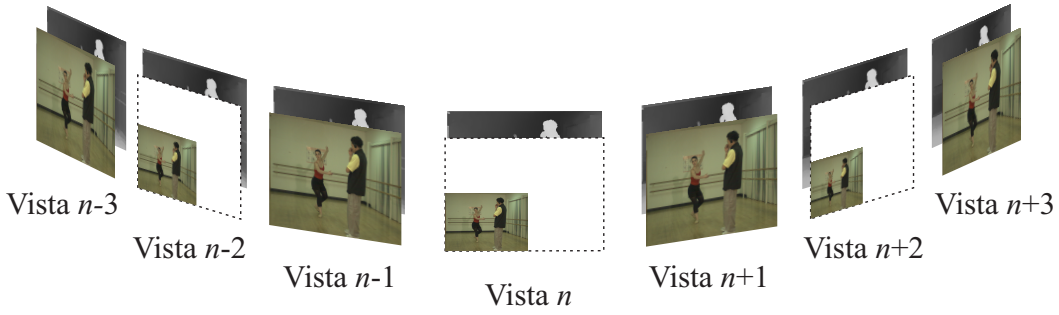


Figura 1.4: Seqüência de vídeo com múltiplas vistas em resolução mista com mapas de profundidade.

Figura 1.5: Seqüência de vídeo com qualidade mista.

1.2 DEFINIÇÃO DO PROBLEMA

Nesta tese, algumas técnicas de realce baseado em exemplos foram propostos para vídeos com qualidade e resolução temporalmente mista. Estas ferramentas permitem que quadros ou imagens com qualidade ou resolução que sejam maiores que a do vídeo alvo, denominados de quadros-chave, sejam utilizados para compor dicionários contendo informações (exemplos) para realçar os quadros do vídeo

com resolução ou qualidade menor, denominados de não-chave. As arquiteturas de qualidade e resolução mista podem ser adotadas em vários cenários, como por exemplo:

- Redução da complexidade de compressão - em vídeos de resolução mista, onde quadros-chave (com resolução maior) são misturados a quadros-não-chave em um vídeo. Observamos que a complexidade do codificador pode ser reduzida ou escalonada, pois reduzimos o número de operações computacionalmente demoradas em quadros menores [15]. No decodificador, um realce pode ser aplicado para que a resolução dos quadros-não-chave aumente. Este processo é denominado na literatura como semi-super-resolução ou super-resolução[2] baseada em exemplos [11, 16, 17]. Uma outra possibilidade de se reduzir a complexidade de compressão seria utilizando qualidade de compressão mista [14]. Entretanto, seu uso se restringe a não exploração de correlação temporal no codificador. O decodificador por sua vez, utiliza-se da correlação temporal para realizar o realce do vídeo.

- Redução da taxa de transmissão - no caso de utilizarmos uma codificação de vídeo com qualidade de compressão mista, podemos afirmar que os quadros de qualidade reduzida possuem uma representação na codificação menor comparada com a codificação com qualidade fixa (com a mesma qualidade dos quadros-chave). Entretanto, a qualidade do vídeo na decodificação também é reduzida. Todavia, podemos aumentar a qualidade dos quadros-não-chave baseados nos quadros-chave resultando em uma relação taxa-distorção ligeiramente melhor que utilizando qualidade fixa [14]. Podemos utilizar o realce para aumentar a qualidade de um vídeo que passa por um canal com capacidade variável, já que grande parte dos controladores de taxa variam a quantização para controlar a quantidade de bits utilizado na codificação [18–20].

- Melhorias na qualidade geral do vídeo baseadas em fotos - em algumas câmeras de vídeo é possível capturar imagens (fotos) durante o processo de filmagem. As câmeras geralmente capturam as fotos com uma resolução maior que os quadros da filmagem. Ao aplicar a super-resolução neste vídeo podemos extrapolar algumas limitações que o sensor de captura de vídeo possui, pois podemos obter após o processo de realce uma resolução de vídeo maior. Este cenário também poderia ser aplicado a princípio com qualquer fotografia para melhorar o vídeo, mas aumentamos o risco de inserirmos informações espúrias durante o realce do vídeo [21].

[2]Nesta tese, o termo super-resolução será utilizado especificamente para se referir ao realce que tem por objetivo o aumento da resolução de um quadro ou imagem.

- Correção de erros de transmissão - existem alguns cenários onde um mecanismo de multi-resolução é utilizado como informação redundante em transmissões que geram pequenos aumentos na taxa. Permitindo comunicações de vídeo confiáveis e com baixa latência em canais muito ruidosos. Para tanto, um vídeo é transmitido juntamente com uma versão de baixa resolução em outro canal mais confiável. Em caso de perdas dos pacotes de transmissão, a informação de baixa resolução é inserida na informação perdida e em seguida é feita a super-resolução [22]. Esta aplicação é citada como um possível cenário de utilização da ferramenta de super-resolução proposta, mas não será tratada nesta tese.

- Correção de foco - em algumas câmeras, quadros com diferentes focos podem ocorrer devido ao atraso nos componentes mecânicos durante a atuação do autofoco. Resultando em uma seqüência de vídeo com foco misto, onde alguns quadros estão com o foco normal enquanto outros estão desfocados. Nesta tese o realce baseado em exemplos é aplicado nos quadros desfocados de forma a aumentar a sua qualidade objetiva e subjetiva.

Os cenários de aplicação testados nesta tese são: (i) vídeo com resolução mista, (ii) vídeo com fotografias durante a gravação, (iii) vídeos com múltiplas vistas em resolução mista com informação de profundidade, (iv) vídeo com qualidade mista.

1.3 MÉTODOS PROPOSTOS

Nesta tese foram propostas algumas técnicas que permitem realçar a qualidade (objetiva e subjetiva) na decodificação de vídeo. Estas técnicas baseiam-se no uso de exemplos, também denominado quadros-chave que são imagens ou quadros com qualidade ou resolução que sejam maiores que a do vídeo alvo. Aos quadros que irão passar pelo processo de realce denominamos de quadros-não-chave. Neste caso, serão compostos dicionários contendo informações (exemplos) dos quadros-chave para super-resolver, ou realçar, os quadros do vídeo com baixa resolução ou qualidade, denominados de não-chaves. As arquiteturas de qualidade e resolução mista podem ser adotadas em vários cenários como: a redução da complexidade durante o processo de compressão, redução da taxa de transmissão, melhorias na qualidade geral do vídeo baseadas em outros quadros ou imagens, correção de erros de transmissão, correção de foco, etc. Nesta tese são propostas duas novas técnicas utilizadas no processo de super-resolução baseada em exemplos: compensação de movimento utilizando blocos multi-escala sobrepostos e a combinação das

informações de múltiplos dicionários. Um novo processo de extração de informação para aplicação em super-resolução utilizando o domínio transformado (DCT, do inglês, *discrete cossine transform*) também é proposto na tese. Por fim, propõe-se uma generalização do processo de realce baseado em exemplos para aplicação em vídeos com variação de qualidade entre quadros. Dentre as possíveis variações de qualidade foram contemplados: parâmetros de quantização (definindo a qualidade da compressão), foco ou ruído.

1.4 APRESENTAÇÃO DO MANUSCRITO

A presente tese é composta por seis capítulos. No Capítulo 2 são tratados conceitos básicos sobre compressão de imagens e vídeo que abordam de maneira sucinta grande parte do embasamento teórico necessário para o entendimento da tese. Além disso, uma revisão bibliográfica sobre redimensionamento de imagens e super-resolução servirão de base para o entendimento do sistema de resolução e qualidade mistas, bem como o processo de realce proposto na tese. Como o tema da tese abrange uma gama enorme de assuntos, optou-se por utilizar uma descrição relativamente superficial dos padrões e das técnicas utilizadas para compressão de imagens e vídeos, sendo indicado ainda leituras complementares das referências. Em seguida, o Capítulo 3 descreve com detalhes algumas das técnicas propostas nesta tese, onde são descritas as ferramentas desenvolvidas para o realce (de resolução e qualidade) baseado em exemplos aplicados em vídeos, como o uso de múltiplos dicionários e a compensação de blocos de tamanhos variados com sobreposição. Em sua seção de experimentos foram realçados vídeos com resolução mista e vídeos com fotografias durante a gravação. No Capítulo 4 um novo método de realce de resolução aplicado no domínio transformado é introduzido, cujos experimentos são realizados em vídeos de resolução mista e vídeos com múltiplas vistas em resolução mista com informação de profundidade. No capítulo 5 uma generalização do processo de super-resolução baseado em exemplos é proposto para realçar diversos vídeos com qualidade mista. Neste caso, são realçados vídeos contendo quadros com: qualidade de compressão mista, foco misto e ruído misto. Finalmente, as conclusões e os trabalhos futuros são descritos no Capítulo 6.

2 REVISÃO BIBLIOGRÁFICA

2.1 COMPRESSÃO DE IMAGENS E VÍDEOS

Nesta seção será apresentada a representação de um vídeo digital em diferentes espaços de cores, seguida de uma pequena descrição sobre codificação sem perdas e codificação com perdas. Em seguida alguns padrões de codificação de imagem e vídeo serão superficialmente descritos de forma a introduzir os conceitos básicos para o entendimento desta tese.

2.1.1 Espaço de cores YUV e amostragem 4:2:0

No caso de imagens digitais monocromáticas, os pixels não carregam a informação de cor. Portanto, as imagens possuem intensidades que visualmente variam entre o branco e o preto, passando por uma escala de cinza.

Já as imagens coloridas carregam as informações de cores. Em uma representação utilizando o espaço de cores RGB (*red*, *green*, *blue*) cada pixel é representado por três componentes. Entretanto, o sistema visual humano tende a perceber o conteúdo das cenas com maior sensibilidade espacial aos detalhes de brilho comparados aos de cor [23]. Isso faz com que o sistema de cores RGB não seja o espaço mais eficiente de representação de cores para a compressão. De posse desta informação, os sistemas de compressão de imagem e vídeo foram desenvolvidos de forma a tirar vantagem desta característica. Os *codecs* de vídeo mais populares [24–31] (como o JPEG, JPEG 2000, MPEG-1, MPEG-2, H.263, MPEG-4 e o H.264) geralmente utilizam o espaço de cor YUV juntamente com uma redução da resolução (sub-amostragem) da informação de crominância U e V. A componente Y é denominada luminância, e representa o brilho. As outras duas componentes U e V representam o quanto a cor se distancia do eixo de luminância (escala de cinza) nos eixos azul e vermelho, respectivamente.

Como dito anteriormente, o sistema visual humano é espacialmente mais sensível à resolução da luminância do que da crominância. Por isso, o MPEG-1, MPEG-2 e H.264 na maioria dos perfis (que serão detalhados na seção 2.1.7.7) utilizam-se de uma estrutura de amostragem onde cada componente de crominância possui um quarto do número de amostras da componente de luminância (mais precisamente, metade do número de componentes tanto na horizontal quanto na vertical). Isso é denominado na literatura

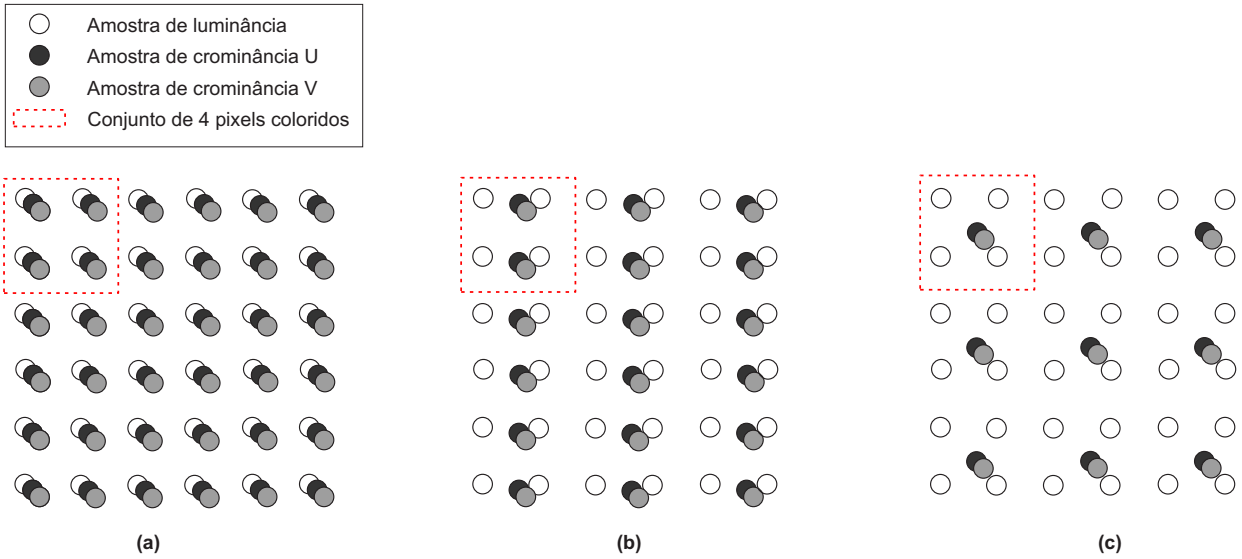


Figura 2.1: Amostras de luminância e crominância nos formatos (a) 4:4:4, (b) 4:2:2 e (c) 4:2:0.

como um vídeo de formato YUV 4:2:0 (Figura 2.1(c)). Tipicamente, são usados oito bits de precisão por amostra. Outras propostas de extensão do padrão do H.264 (denominada de FREXt, do inglês, *fidelity range extension*) para suportar vídeos de alta fidelidade permitem uma maior resolução de crominância e maior precisão de bits por amostra. A Figura 2.1(a) mostra o caso onde para 4 pixels existem 4 componentes de luminância, 4 de crominância U e 4 de crominância V, conhecido na literatura como formato YUV 4:4:4. Outro formato similar é o YUV 4:2:2, que segue a mesma lógica do caso anterior, conforme mostra a Figura 2.1(b).

2.1.2 Compressão de vídeo digital

A compressão de vídeo digital, para a teoria da informação, é um tipo de codificação de fonte, onde a quantidade de bits é reduzida para sua representação [5]. Logo, o objetivo principal da compressão é representar um sinal com o menor número de bits possível, com uma "qualidade aceitável" que depende da aplicação. Existem basicamente dois tipos de compressão:

- Sem perdas, também conhecido como codificação de entropia, onde obtemos uma reconstrução perfeita do sinal original. Para tanto, são exploradas as estatísticas do sinal.

- Com perdas, este caso é caracterizado por haver uma perda irreversível de informação. Portanto, apenas uma aproximação do sinal original poderá ser reconstruída a partir do sinal codificado.

2.1.2.1 Compressão sem perdas ou codificação de entropia

O codificador de entropia é um tipo de compressor de informação sem perdas, extremamente dependente do modelo de probabilidades empregado à informação. A codificação de entropia se utiliza de conceitos básicos da teoria de informação ou teoria estatística das comunicações [32], cuja fonte de informação pode ser vista como sendo um processo que gera uma seqüência de símbolos pertencentes a um alfabeto finito. Neste contexto, o codificador converte uma série de símbolos que representam dados, pixels ou elementos da seqüência de vídeo (coeficientes de transformada quantizados, vetores de movimento, cabeçalhos e informações suplementares) em uma seqüência de bits comprimida sem perda de informação.

Os modelos de fontes de informação mais comuns em codificadores de vídeo são: fontes discreta sem-memória (DMS, do inglês, *discrete memoryless source*) e fontes de Markov. Os códigos de tamanho variável (VLC, do inglês, *variable length coding*) são baseados no modelo DMS, e códigos preditivos são baseados no modelo Markoviano [5].

No caso das fontes DMS, cada símbolo da fonte é gerado independentemente, tornando-os estatisticamente independentes. A fonte é completamente definida pela relação entre símbolos (ou eventos) e pela probabilidade de ocorrência de cada símbolo, sendo $E = \{e_1, e_2, ..., e_n\}$ os símbolos, $P = \{\rho(e_1), \rho(e_2), ..., \rho(e_n)\}$ o conjunto de probabilidades de ocorrência de cada símbolo e n o número de símbolos do alfabeto.

Outro conceito importante é o da entropia, que é definido como sendo a média da autoinformação, ou seja da informação contida na fonte. A autoinformação é definida por:

$$I(e_i) = \log_b \left\{ \frac{1}{\rho(e_i)} \right\}, \tag{2.1}$$

onde a base do logaritmo b é determinada pelo número de estados utilizados para representar a informação da fonte. Ou seja, para fontes de informações digitais utiliza-se a base $b = 2$ para se obter o conteúdo da informação em bits por símbolo, ou taxa de bits. E $\rho(e_i)$ é a probabilidade de ocorrência do evento ou símbolo e_i. A entropia da fonte é definida por [32]

$$\begin{aligned} H(E) &= \sum_{i=1}^{n} \rho(e_i) I(e_i) \\ &= -\sum_{i=1}^{n} \rho(e_i) \log_b \{\rho(e_i)\}. \end{aligned} \tag{2.2}$$

A entropia quantifica a média de bits por símbolo necessária para representar a informação contida na fonte. O teorema de codificação de fontes sem ruído afirma que uma fonte pode ser codificada com uma média de bits por símbolo muito próxima, mas não menor que a entropia da fonte. Logo, os codificadores de entropia procuram utilizar códigos cujo desempenho se aproxime da entropia da fonte.

Duas técnicas de codificação de entropia são amplamente usadas: a codificação de comprimento variável e a codificação aritmética. A codificação de comprimento variável utiliza códigos instantâneos que consistem em mapear os símbolos que entram no codificador de entropia em uma série de palavras-código de comprimento variável. Logo, cada uma delas deve conter um número inteiro de bits. Os símbolos com maior freqüência de ocorrência são representados com códigos menores e o contrário ocorre com símbolos de menor freqüência.

No caso do JPEG, o código de Huffman [33] é utilizado para sua compressão entrópica. Apesar de ter um desempenho satisfatório, a codificação de comprimento variável possui a limitação de associar uma quantidade inteira de bits a um símbolo - impedindo a compressão de se aproximar da entropia [5]. Por isso, a codificação aritmética torna-se uma alternativa à codificação de comprimento variável, dado que, com ela, é possível que se chegue mais perto das taxas de compressão teóricas [3, 34], visto que um codificador aritmético é capaz de converter uma seqüência de símbolos de dados em um único símbolo codificado.

Como a eficiência da codificação de entropia depende, em grande parte, do modelo de probabilidade de ocorrência de símbolos usados, a codificação aritmética adaptativa baseada em contexto [35], que usa características locais para estimar a probabilidade de ocorrência de cada símbolo a ser codificado, se torna um dos codificadores de entropia de melhor desempenho a ser utilizado atualmente, sendo inclusive adotado em vários codificadores de imagens e vídeos como o JBIG2 [36], JPEG2000 [25] e o H.264 [31].

Maiores detalhes sobre a codificação de entropia não serão tratados nesta tese, sendo recomendada a consulta de trabalhos como [5, 37].

2.1.2.2 Codificação com perdas

As técnicas de codificação com perdas são mais utilizadas por conseguirem reduções na taxa de bits que tipicamente são menores que as das codificações sem perdas em algumas ordens de grandeza. No entanto, grande parte dos padrões de codificação com perdas possui uma etapa de codificação de entropia. Os algoritmos de compressão de vídeo exploram algumas características inerentes ao vídeo, como altas correlações espaciais e temporais.

Na compressão de vídeo, utilizam-se técnicas de compressão de imagens para explorar a correlação espacial de um quadro. Aplica-se uma transformada (por sub-bandas ou em blocos - de modo a compactar a energia do sinal), uma quantização (esse parâmetro controla a qualidade e por conseguinte, a taxa da compressão), e por fim aplica-se um codificador de entropia.

Mas para explorar a redundância temporal, utilizamos as técnicas de estimação de movimento, que consiste em buscar em um quadro previamente codificado (denominado na literatura como quadro de referência) um bloco de informação equivalente àquela do quadro que está sendo codificado. Este processo estima o movimento translacional que o conteúdo dos blocos realizaram entre um quadro e outro, do qual resultam os vetores de movimento. A compensação de movimento é a aplicação do vetor de movimento no quadro de referência, de forma que uma predição desta informação possa ser obtida. Esse processo resulta em um novo bloco ou quadro estimado a partir de um quadro codificado anteriormente, onde para a reconstrução completa deste quadro são necessários: o(s) quadro(s) de referência, os vetores de movimento, a descrição dos particionamentos dos blocos (se houver) e o resíduo - que é a diferença entre o bloco ou quadro a ser comprimido e o estimado. Os resíduos são codificados utilizando as mesmas técnicas utilizadas para explorar a correlação espacial.

A maioria dos padrões de codificação de vídeo utiliza a mesma estrutura básica [5], denominada de codificação híbrida, onde a codificação por transformada é utilizada em conjunto com o DPCM (do inglês, *differential pulse code modulation*), representada pela codificação por compensação de movimento.

2.1.3 Padrão JPEG

Para se comprimir o primeiro quadro de um vídeo, utilizam-se técnicas similares às técnicas de compressão de imagens, como o JPEG. A compreensão das técnicas contidas neste padrão facilita o estudo de codificadores mais modernos. O JPEG é um padrão de codificação baseado em blocos (do inglês, *block based*) e no caso, divide a imagem em blocos de 8×8 pixels. Se a imagem não possui o número de linhas e número de colunas múltiplo de 8, um preenchimendo na imagem é feito para se obter essa característica. Um exemplo dessa divisão pode ser visto na Figura 2.2:

A Equação 2.3 mostra a matriz $\mathbf{B}_p$ que representa um bloco de pixels dessa imagem que será usada como exemplo. Em cada bloco 8×8, são feitas uma série de operações, exemplificada pela matriz $\mathbf{B}_p$ a

seguir:

$$\mathbf{B}_p = \begin{bmatrix} 104 & 108 & 107 & 101 & 94 & 95 & 98 & 102 \\ 96 & 100 & 103 & 100 & 96 & 74 & 75 & 73 \\ 77 & 69 & 70 & 87 & 84 & 64 & 64 & 67 \\ 71 & 60 & 52 & 59 & 64 & 56 & 54 & 57 \\ 58 & 53 & 51 & 54 & 52 & 51 & 52 & 52 \\ 53 & 50 & 53 & 52 & 52 & 58 & 51 & 47 \\ 48 & 53 & 53 & 51 & 53 & 55 & 51 & 53 \\ 47 & 48 & 48 & 47 & 55 & 47 & 51 & 48 \end{bmatrix}. \tag{2.3}$$

Inicialmente, a imagem é subtraída de 128 (no caso de imagens de 8 bits), de forma que os pixels variem entre -128 e 127. Para cada bloco, a DCT é aplicada, de forma a decompor os valores de intensidade do bloco em bases senoidais, como ilustra a Figura 2.3, resultando nos coeficientes mostrados na matriz $\mathbf{B}_t$ da Equação 2.4, que são ponderações das bases mostradas na Figura 2.3. Note que os coeficientes mais próximos do topo, à esquerda, são os coeficientes das freqüências mais baixas.

$$\mathbf{B}_t = \begin{bmatrix} -495,5000 & 19,8185 & -7,8394 & 0,0189 & 10,0000 & -0,9774 & -3,2472 & 2,7423 \\ 134,5957 & 21,6732 & -2,7443 & -8,5780 & 7,3955 & 1,4963 & 3,0981 & 0,2423 \\ 58,8497 & 1,3480 & -0,8839 & -10,4132 & -9,1595 & -2,9341 & -0,6339 & 2,9325 \\ 17,4016 & -3,4000 & 8,5789 & -2,6026 & -13,8331 & 1,2787 & 6,4987 & -4,3676 \\ -4,5000 & -6,9291 & 14,3858 & 2,8536 & -1,5000 & -0,3074 & -0,5468 & -0,0110 \\ 2,1127 & -10,2568 & 7,1211 & 3,3039 & 0,4213 & -1,9182 & 2,3304 & -4,1175 \\ -1,6461 & -9,0033 & -1,1339 & 3,2009 & 2,7116 & 3,4405 & 0,8839 & -1,5484 \\ 0,6826 & -6,7326 & -0,4247 & -4,3075 & 2,4544 & 1,5464 & -1,1040 & -2,1524 \end{bmatrix} \tag{2.4}$$

Em seguida, os coeficientes da DCT são quantizados, segundo uma matriz de quantização. Uma das matrizes de quantização utilizada pelo JPEG é mostrada na Equação 2.6, que foi desenvolvida empiricamente, mas direcionada pelo sistema visual humano (HVS, do inglês, *human visual system*). Os passos de quantização utilizados para definir a qualidade de uma imagem utilizam valores proporcionais à matriz da Equação 2.6. Os coeficientes da DCT são quantizados dividindo-se cada elemento da matriz de coeficientes pelo elemento correspondente da matriz de quantização, em seguida adicionando-se $0,5$ e truncando-se o resultado, conforme descreve a Equação 2.5:

$$B_{tq}(i,j) = \left\lfloor \left(\frac{B_t(i,j)}{Q(i,j)} + 0,5 \right) \right\rfloor \tag{2.5}$$

Figura 2.2: Imagem dividida em blocos de 8×8 pixels.

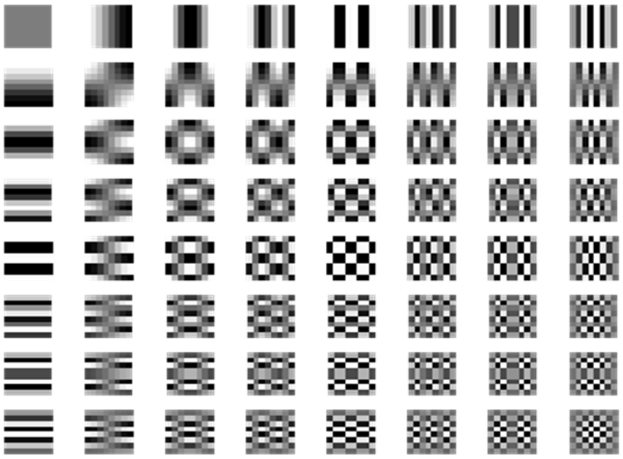


Figura 2.3: Bases da DCT.

onde B_{tq} são os coeficientes transformados e quantizados, $\lfloor \cdot \rfloor$ é a operação de truncamento para o maior valor inteiro menor que o argumento, B_t são os coeficientes transformados, Q são os coeficientes da matriz de quantização e (i, j) são as posições dos elementos na matriz

$$\mathbf{Q} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}. \tag{2.6}$$

Existem várias matrizes de quantização possíveis no formato JPEG. Cada uma delas gera uma relação diferente entre a compressão e a qualidade desejada. Por exemplo, se fosse utilizada uma matriz de quantização unitária (denominador da Equação 2.5), obter-se-ia uma maior qualidade, porém uma menor compressão. Observe que na matriz de quantização utilizada, os coeficientes ficam maiores à medida que se afastam da posição do coeficiente DC (topo, à esquerda), indicando que as freqüências mais baixas são priorizadas.

$$\mathbf{B}_{tq} = \begin{bmatrix} -31 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 11 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.7}$$

Os coeficientes quantizados são mostrados na matriz $\mathbf{B}_{tq}$ da Equação 2.7. Note que os coeficientes quantizados possuem uma grande quantidade de zeros nas posições referentes às maiores freqüências. Isto ocorre quando o conteúdo do bloco possui pouca variação espacial ou quando utilizamos quantizadores que possuem maior intensidade nas componentes de alta freqüência. Para tirar proveito desta característica,

percorre-se os coeficientes da matriz quantizada em ziguezague, conforme mostra a Figura 2.4, obtendo-se um vetor no final desse processo.

Isto implica a geração de seqüência de coeficientes com valores iguais, permitindo-se comprimir com eficiência utilizando a técnica de *run-length encoding*, onde um segmento do vetor de coeficientes que possui valores iguais (*run*) são descritos tipicamente por dois bytes, o primeiro pela quantidade de coeficientes no *run* e o segundo pelo valor dos coeficientes no *run*.

Além disso, a utilização de códigos especiais, como *End of Block*, que indica que todos os coeficientes seguintes são nulos - o que implica em um grande aumento da eficiência do ziguezague. Finalmente, o padrão JPEG utiliza o código de Huffman, com uma tabela específica, que associa o valor dos coeficientes a um código, conhecida tanto pelo codificador quanto pelo decodificador.

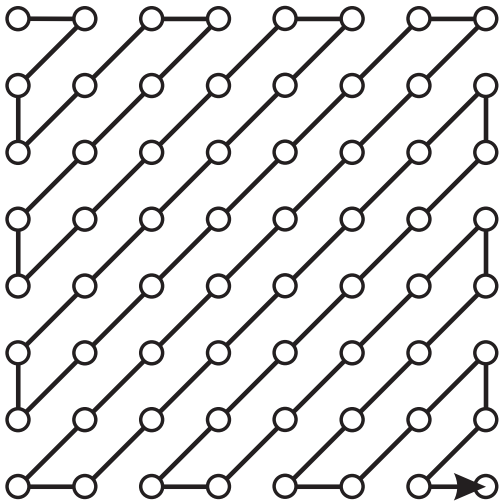


Figura 2.4: Reordenação dos coeficientes da DCT utilizando ziguezague.

A descompressão é simples: decodifica-se o código de Huffman, o que permite a remontagem das matrizes cujos coeficientes DCT foram quantizados. Nesse ponto, basta re-escalar a matriz, aplicar a transformada discreta inversa do cosseno (IDCT) e somar 128. O resultado pode ser visto na matriz $\mathbf{B}_r$ da Equação 2.8

$$\mathbf{B}_r = \begin{bmatrix} 108 & 108 & 106 & 104 & 100 & 95 & 91 & 85 \\ 96 & 95 & 94 & 92 & 89 & 84 & 81 & 78 \\ 78 & 78 & 77 & 76 & 73 & 69 & 69 & 64 \\ 63 & 64 & 64 & 63 & 61 & 58 & 55 & 53 \\ 55 & 55 & 56 & 56 & 55 & 53 & 51 & 49 \\ 51 & 51 & 53 & 53 & 52 & 52 & 51 & 49 \\ 48 & 49 & 51 & 52 & 53 & 52 & 51 & 50 \\ 47 & 48 & 50 & 51 & 52 & 52 & 51 & 51 \end{bmatrix}. \tag{2.8}$$

A diferença entre o bloco reconstruído e o bloco original é mostrada na matriz $\mathbf{D}_b$ da Equação 2.9. A percepção visual dessa diferença dos pixels é muitas vezes pequena (dependendo do quantizador), fazendo com que o algoritmo apresente bons resultados.

$$\mathbf{D}_b = \begin{bmatrix} -4 & 0 & 1 & -3 & -6 & 0 & 7 & 13 \\ 0 & 5 & 9 & 8 & 7 & 10 & -6 & -5 \\ -1 & -9 & -7 & 11 & 11 & -5 & -2 & 3 \\ 8 & -4 & -12 & -4 & 3 & -2 & -1 & 4 \\ 3 & -2 & -5 & -2 & -3 & -2 & 1 & 3 \\ 2 & -1 & 0 & -1 & -1 & 6 & 0 & -2 \\ 0 & 4 & 2 & -1 & 0 & 3 & 0 & 3 \\ 0 & 0 & -2 & -4 & 3 & -5 & 0 & -3 \end{bmatrix} \tag{2.9}$$

Aplicando a compressão em todos os blocos da Figura 2.2, teremos na Figura 2.5 o resultado da codificação de uma imagem. A Figura 2.5(a) mostra a imagem original, e a Figura 2.5(b) mostram a imagem codificada com a tabela de quantização apresentada na Equação 2.6. A PSNR[1] da imagem reconstruída foi de 36,49 dB. Subjetivamente, os resultados da Figura 2.5 mostra uma imagem comprimida muito próxima à imagem original, mas com uma representação em bits de aproximadamente 12,25 vezes menor que a imagem sem compressão.

[1]A PSNR (*peak signal-to-noise ratio*) é calculada por: $PSNR_{dB} = 10\log_{10}\frac{(2^n-1)^2}{MSE}$, onde MSE é o erro médio quadrático (*mean square error*) entre a imagem reconstruída ($\mathbf{I}_r$) e a imagem original ($\mathbf{I}_o$) de tamanhos $w \times h$, dado por $MSE = \frac{1}{wh}\sum_{i=1}^{w}\sum_{j=1}^{h}[\mathbf{I}_r(i,j) - \mathbf{I}_o(i,j)]^2$ e n o número de bits de precisão de um pixel. No caso de processamento de vídeo, tipicamente, utiliza-se a média das PSNRs da luminância dos quadros.

(a) (b)

Figura 2.5: (a) Imagem original. (b) Reconstrução da imagem comprimida com JPEG com PSNR de 36,49 dB.

2.1.4 *Motion* JPEG e *Motion* JPEG 2000

O Motion JPEG é um *codec* de vídeo que faz o uso da técnica de compressão de imagens do JPEG (descrito na Seção 2.1.3), mas não se utiliza das predições entre quadros [24]. A ausência de predições *inter* quadros resulta em uma ineficiência na capacidade de compressão, por não explorar a correlação, mas facilita a edição do vídeo, uma vez que as edições podem ser realizadas em qualquer quadro. Diferentemente do MPEG-2 e do H.264 cujo conteúdo depende de informação de quadros anteriores e posteriores - o que torna mais difícil a visualização completa do quadro para a edição. Outros compressores como o MPEG-2 e o H.264 quando operados de forma que utilizem apenas quadros *intra* (I) possui facilidades similares para edição.

O Motion JPEG é muito utilizado em circuito fechado de televisão ou sistema de monitoramento por câmeras, cuja aplicação faz uso de pouca resolução temporal (em torno de 5 quadros por segundo). Com a amostragem temporal baixa, geralmente se diminui a correlação temporal, resultando em uma menor eficiência na compressão *inter* quadros. Além disso, como os quadros são totalmente independentes entre si, o sistema é mais robusto a falhas na gravação, transmissão e exibição, haja vista que o erro de um quadro não propaga ou interfere nos demais.

O JPEG 2000 segue uma abordagem um pouco diferente do JPEG, onde para toda imagem a Transformada de Wavelets Discreta (DWT) é aplicada e seus coeficientes, inicialmente organizados em

sub-bandas, são quantizados seguindo técnicas como: EBCOT - *Embedded Block Coding with Optimized Truncation*. Em seguida os coeficientes são comprimidos entropicamente utilizando um codificador aritmético. De maneira análoga, o JPEG 2000 possui uma variante (Parte 3) do padrão de codificação de imagens, que é um *codec* de vídeo, conhecido como Motion-JPEG 2000 [38], cujo núcleo de compressão é baseado no JPEG 2000 Parte 1. Atualmente é utilizado em cinemas digitais, gravação e edição de vídeos de alta qualidade baseados em quadros, armazenamento de vídeo em câmeras digitais, imagens médicas e imagens de satélites.

2.1.5 Predição entre Quadros (Temporal)

Em geral, os quadros vizinhos de uma seqüência de vídeo são muito correlacionados, pois foram capturados em instantes de tempo muito próximos entre si. Por exemplo, se a câmera de vídeo permanecer estática, todo o fundo (*background*) da imagem se mantém, e os únicos pixels diferentes são aqueles que representam objetos que se moveram (ou objetos que apareceram atrás dos objetos que se moveram).

A forma mais adotada para explorar a redundância entre quadros (inteiros ou parte deles) vizinhos é prever o quadro atual a partir dos quadros anteriormente codificados e reconstituídos (localmente decodificados). O método mais simples é comparar o quadro atual ao anterior, calcular a diferença entre eles (chamada de resíduo) e codificar apenas essa diferença. Esta diferença, de maneira geral, apresenta muito menos entropia do que o quadro completo. O resíduo é codificado utilizando-se alguma técnica de compressão espacial, mas, como tem muito menos energia (informação) do que o quadro completo, o resultado da compressão é mais eficiente. Este processo é ilustrado na Figura 2.6.

(a) (b) (c)

Figura 2.6: (a) e (b) Quadros sucessivos. (c) Resíduo entre eles.

2.1.5.1 Estimação de Movimento

Estimação de movimento é o processo realizado para encontrar os movimentos translacionais resultantes que ocorrem entre pelo menos dois quadros. Esse processo consiste em procurar o melhor casamento de uma parte da cena presente num quadro (geralmente chamado atual) em outro quadro (anteriormente codificado e reconstruído) que é o quadro de referência.

Neste método, o quadro atual é dividido em regiões que poderiam ser tão pequenas quanto um pixel, porém torna-se inviável devido ao grande esforço computacional exigido e ao grande número de bits para se descrever o deslocamento dessa região (vetores de movimento) que necessitam ser codificados. A informação relativa ao deslocamento de cada região do quadro de referência é chamada de *optical flow* (fluxo óptico) ou simplesmente vetores de movimento. Canonicamente, estima-se o movimento em blocos da imagem, por isso a técnica é chamada *block-based motion estimation* ou estimação de movimentos baseada em blocos.

Na estimação de movimento (*motion estimation*) procura-se no quadro de referência (que pode estar no passado ou futuro, desde que já tenha sido codificado), a região que melhor representa o bloco atual. Geralmente a área em que se procura o bloco atual no quadro de referência é limitada, pois considera-se que os quadros capturados em um vídeo possuem uma latência tão pequena que os deslocamentos dos objetos em cena são restritos às suas vizinhanças, e seria computacionalmente exaustivo procurar por todo o quadro.

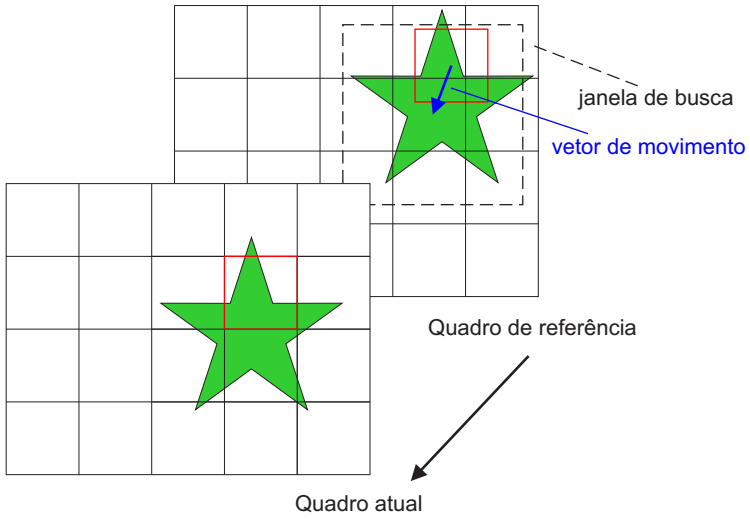


Figura 2.7: Estimação de movimento.

O critério de avaliação é tipicamente feito minimizando a SAD (*sum of absolute differences*, soma das diferenças absolutas) ou a SSD (*sum of square differences*, soma das diferenças quadráticas). A Figura 2.7, mostra dois quadros: o atual (a ser codificado) e o de referência (reconstruído). O deslocamento relativo de um bloco no quadro atual são geralmente calculados pela Equação 2.10 ou 2.11:

$$D_{SAD} = \sum_{i=1}^{N} \sum_{j=1}^{N} \left| p_{pixel\ atual}(i,j) - p_{pixel\ ref}(i+x, j+y) \right|, \tag{2.10}$$

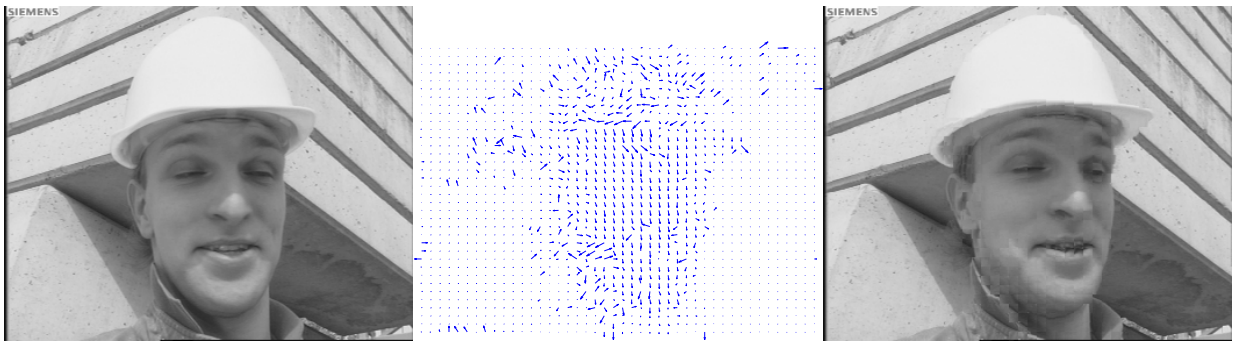
$$D_{SSD} = \sum_{i=1}^{N} \sum_{j=1}^{N} \left(p_{pixel\ atual}(i,j) - p_{pixel\ ref}(i+x, j+y) \right)^2, \tag{2.11}$$

onde x e y representam os deslocamentos (vetores de movimento) no processo de busca pelo bloco no quadro anteriormente codificado (referência). Para o cálculo de vetores de movimento variam-se os valores de (x, y) em torno de uma vizinhança ou no quadro inteiro de forma a minimizar a D_{SAD}, D_{SSD} ou qualquer outro critério escolhido. No caso mostrado na Figura 2.7, a procura é feita dentro de uma janela de busca de tamanho $m \times n$ pixels. O deslocamento da procura no quadro anterior pode ser definido utilizando-se diversos algoritmos de buscas como: espiral, circular, hexagonal, telescópica, diamante, completa, EPZS, etc [39–44].

2.1.5.2 Compensação de Movimento

A compensação de movimento seria a aplicação dos vetores de movimento nos quadros de referência de modo a gerar uma predição do quadro (ou parte do quadro) atual. O decodificador deve usar os vetores de movimento para criar o quadro compensado, decodificar o resíduo, e utilizar os dois para formar o quadro final que será exibido. A Figura 2.8(a) apresenta um quadro de referência previamente codificado e reconstruído, já a Figura 2.8(b) mostra o fluxo óptico entre quadros subseqüentes. Ao se aplicar os vetores de movimento no quadro de referência (isto é, realizar a compensação de movimento) obtém se o quadro compensado da Figura 2.8(c).

O uso de blocos retangulares é muito popular, mas tem algumas desvantagens. Objetos reais em geral têm bordas mais complexas, que não acompanham as bordas retangulares dos blocos usados para compensação de movimento. Em trabalhos como [45–47] foram explorados alguns formatos arbitrários



(a) (b) (c)

Figura 2.8: (a) Quadro de referência; (b) Fluxo óptico (c) Quadro compensado (estimado).

para compensação de movimento, obtendo ganhos significativos na compressão. Além disso, movimentos mais complexos como: zoom, rotações, deformações, torções, oclusões, são difíceis de se estimar. Apesar disso, o fato de ser computacionalmente viável e ser compatível com transformadas baseadas em blocos, como a DCT, fez essa técnica ser utilizada por quase todos padrões de compressão de vídeo.

A compensação de movimento é utilizada no codificador durante a reconstrução do quadro de referência, de forma a sincronizar as informações com o decodificador. Ou seja, como o sistema de compressão em questão gera perdas, faz-se necessário que o codificador tenha as mesmas referências do decodificador para que o erro entre o quadro original e o com perdas não se propague. A compensação de movimento é um dos elementos chave no processo de decodificação de um vídeo e é responsável pela predição temporal dos quadros, já que os *codecs* geralmente operam baseados no DPCM, onde se faz a predição do sinal a ser codificado e se codifica apenas sua diferença.

2.1.6 MPEG-1 e MPEG-2

Atualmente, o compressor de vídeo mais popular, utilizado por muitas operadores de TV digital, nos DVDs e em *streaming* de vídeo na internet, se baseia no JPEG (o que pode ser facilmente observado pelas operações de transformada, quantização e codificação de entropia no diagrama de blocos da Figura 2.9) e foi desenvolvido por um grupo denominado *Motion Picture Experts Group* popularmente conhecido como MPEG.

No caso do MPEG-1 [26], o desenvolvimento foi otimizado para aplicações que utilizem aproximadamente 1,5 Mbps com um vídeo com resolução SIF (320×240 pixels) a trinta quadros por segundo. Já o MPEG-2 [27] foi proposto para atender codificações de vídeo de alta qualidade com taxas superiores

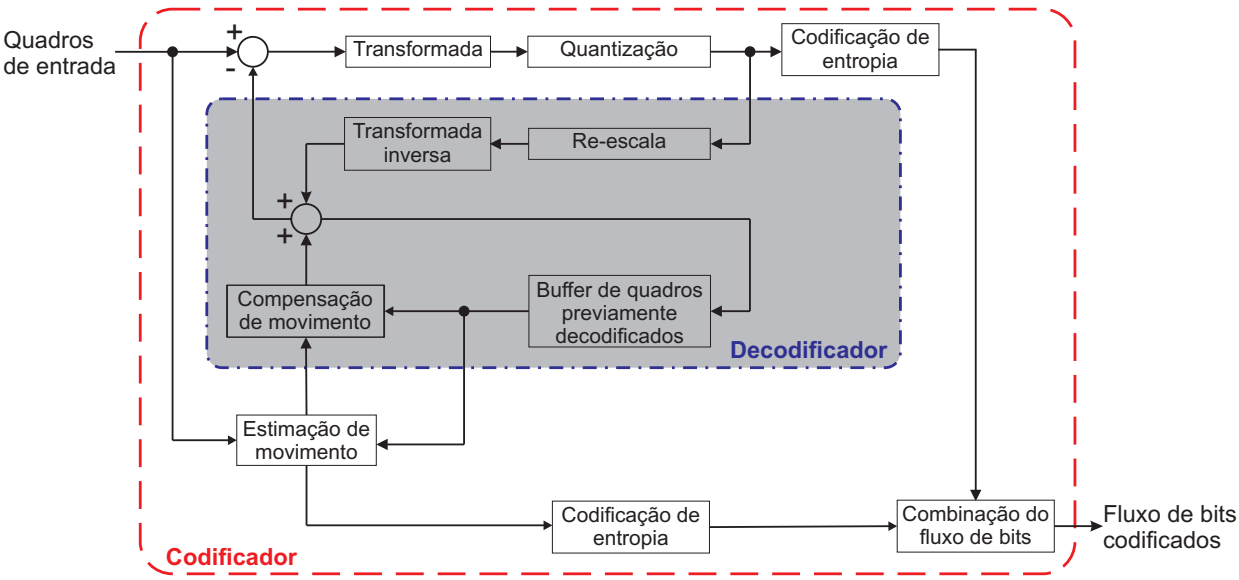


Figura 2.9: Diagrama de blocos simplificado do MPEG-1 e MPEG-2.

a 1,5 Mbps para compressão de vídeos com resolução SD (720×576 pixels). O desenvolvimento do MPEG-2 foi fortemente baseado no MPEG-1, tanto que existem alguns modos que são compatíveis entre si. Ou seja, o MPEG-2 consegue decodificar um fluxo de bits de um vídeo comprimido com o MPEG-1, o que induz a afirmar que o MPEG-2 é uma adição de propriedades e características no MPEG-1, tal como: suporte tanto a vídeo progressivo (Figura 2.10(a)) quanto entrelaçado (Figura 2.10(b)), sendo que os quadros entrelaçados são muito utilizados para evitar erros de *flicker* (onde a seqüência de vídeo "pisca" devido à falta de atualização da imagem); vetores de movimento com meio pixel de precisão (Figura 2.11); escalabilidade de qualidade (SNR, do inglês, *signal to noise ratio*); escalabilidade espacial; escalabilidade temporal; permite ainda o particionamento de dados, que garante maior robustez contra erros e perdas de pacotes; e ocultamento de erros, que é um mecanismo de minimização de erros de transmissão no decodificador.

Os *codecs* MPEG-1 e MPEG-2 utilizam o GOP do tipo IBBP isto significa que a primeira imagem é comprimida como um quadro *intra* ao utilizar os blocos de transformada, quantização e codificação de entropia mostrados na Figura 2.9. Ou seja, esta codificação é baseada apenas nas informações contidas no quadro, e que em seguida um novo quadro é comprimido com base no quadro anteriormente comprimido (que é localmente decodificado e armazenado no *buffer* de quadros previamente codificados, conforme mostra a Figura 2.9), o que caracteriza um quadro do tipo P ou predito.

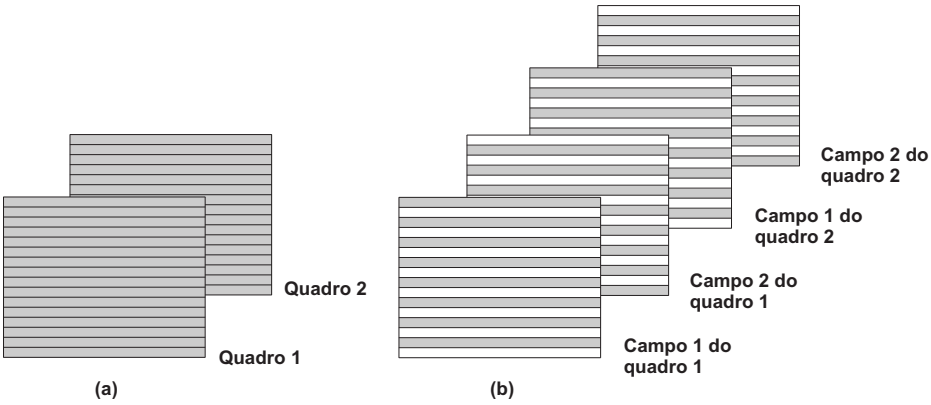


Figura 2.10: (a) Varredura de quadro progressiva; (b) Varredura de quadro entrelaçada.

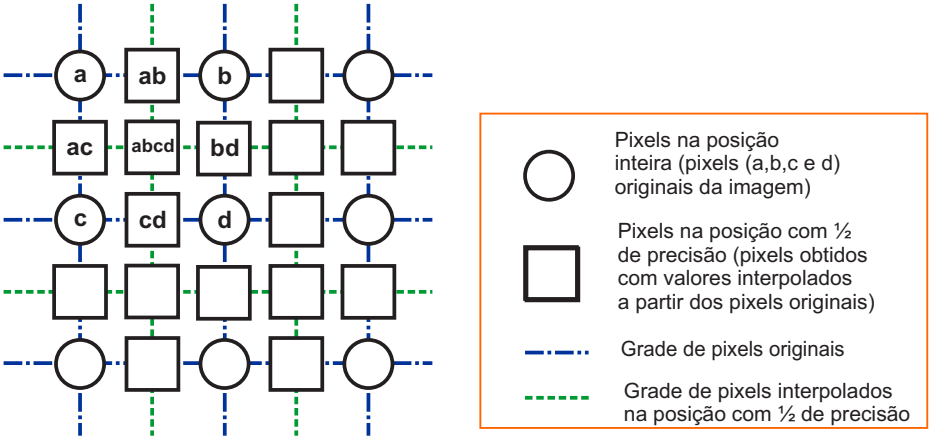


Figura 2.11: Ilustração de um bloco com precisão de meio pixel para estimação de movimento.

O quadro P é codificado a partir da predição do quadro atual ao estimar o movimento em relação ao quadro anteriormente codificado, e em seguida o resíduo (diferença entre o quadro estimado e o quadro original) é comprimido utilizando a codificação similar àquela dos quadros *intra*.

Já os quadros do tipo B ou biprevistos são estimados e compensados a partir dos quadros do tipo I e do tipo P, previamente codificados e reconstruídos, ou seja, localmente decodificados, de forma que o codificador tenha a mesma informação que o decodificador para gerar as mesmas predições. Os resíduos são então codificados de modo similar aos quadros *intra*.

Observe a ilustração das Figuras 2.12 e 2.13. Note que a ordem de compressão dos quadros é diferente da seqüência de exibição do mesmo. Isto permite a compensação de quadros baseados em quadros "futuros" tendo como parâmetro a seqüência de exibição (Figura 2.12).

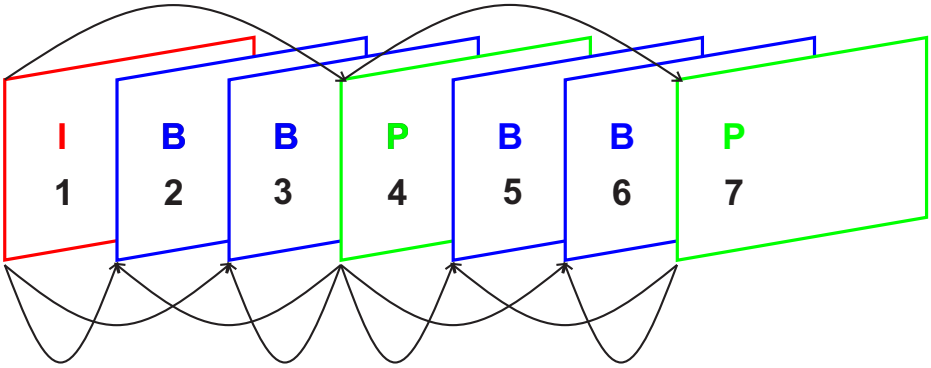


Figura 2.12: GOP típico de uma compressão MPEG-2, ordenado de acordo com a seqüência de exibição.

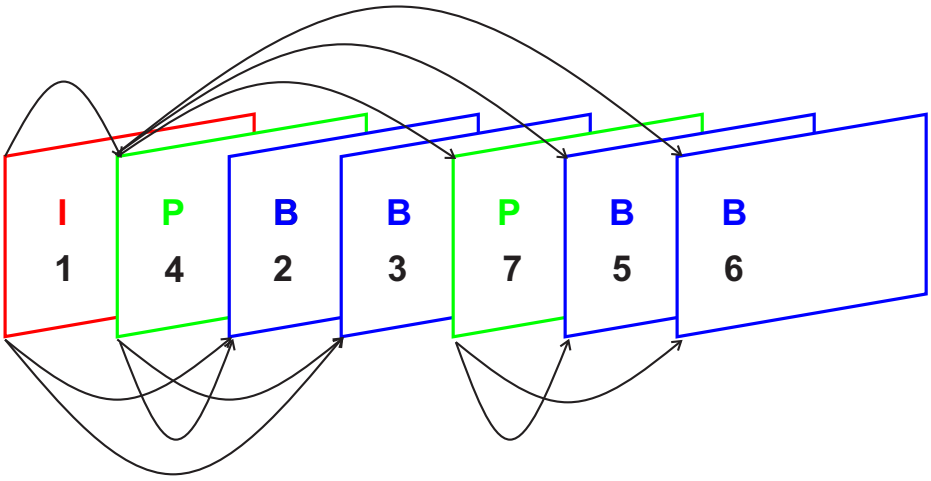


Figura 2.13: GOP típico de uma compressão MPEG-2, ordenado de acordo com a seqüência de compressão.

No MPEG os quadros são constituídos de *slices*, que seriam conjuntos de macroblocos contíguos em ordem lexicográfica (*raster scan*), exemplificada na Figura 2.14. Os *slices* são importantes para aumentar a robustez do sistema contra erros no canal. Por exemplo, se um fluxo de bits contiver um erro em um bit, o erro causaria uma propagação devido à codificação de tamanho variável. No entanto, no próximo *slice* uma ressincronização do decodificador ocorre. Fazendo com que apenas o *slice* com erro de transmissão seja descartado ou retransmitido.

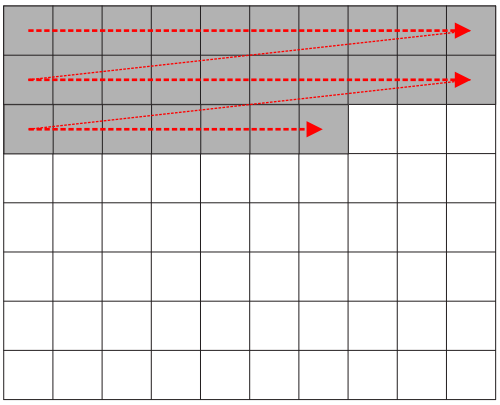


Figura 2.14: Exemplo de um *slice* em um quadro comprimido com MPEG.

Existem quatro tipos de macroblocos no MPEG-2: (a) *intra*; (b) *inter* predito pelo quadro anterior ou posterior; (c) *inter* bipreditos por uma média entre informações dos quadros anterior e posterior e (d) *skip*, que é designado quando o vetor de movimento bem como todos seus coeficientes da DCT são nulos.

Como é mostrado na Figura 2.15, existem seis camadas de codificação no fluxo de bits do MPEG-1 e no MPEG-2: seqüência de vídeo, GOP, quadro, *slice*, macrobloco, e bloco.

A camada de uma seqüência de vídeo é basicamente constituída de um cabeçalho, um ou mais GOPs, e um código de fim de seqüência. O conteúdo dela contém uma série de parâmetros como tamanho da figura (dimensão horizontal e vertical em pixels), taxa de quadros por segundo, taxa de bits, tamanho mínimo de *buffer*, etc.

A camada do GOP consiste em um conjunto de quadros dispostos na ordem de exibição, e contém uma série de parâmetros: como o código de tempo, que fornece as horas, minutos e segundos do intervalo de tempo desde o início da seqüência, e *flags* que indicam qual quadro de referência será utilizado pelo decodificador.

A camada de quadros atua como uma unidade primária de codificação, que contém uma série de parâmetros: a referência temporal, que identifica o número do quadro, ou seja, a seqüência para determinar

a ordem de exibição; tipo de quadro (I, P ou B) e a ocupação inicial do *buffer* de decodificação, evitando *overflow* e *underflow*; e ainda os vetores de movimento dos quadros P e B.

Já a camada de *slice* atua como uma unidade ressincronizadora, que contém a posição inicial do *slice* e o fator de quantização pelo qual o referido *slice* foi codificado.

A camada de macroblocos age como uma unidade para compensação de movimentos, e contém os seguintes parâmetros: incremento do endereçamento do macrobloco, tipo de macrobloco, fator de quantização, vetores de movimento, e a forma de se codificar os blocos do macrobloco.

A camada de blocos é a camada de nível mais baixo da seqüência de vídeo e consiste na codificação dos coeficientes da DCT de 8×8. Quando um macrobloco é codificado no modo *intra*, os coeficientes DC de uma imagem são codificados de maneira similar ao JPEG, onde o coeficiente DC do macrobloco atual é predito a partir do coeficiente DC do macrobloco anterior. No início de cada *slice* é atribuído o valor de 1024 para a predição dos coeficientes DC para os blocos de luminância e crominância. Os valores diferenciais de DC são codificados utilizando o VLC para representar a informação residual. Finalmente, os coeficientes AC são codificados utilizando VLC para representar os valores codificados utilizando o *run-length encoding*.

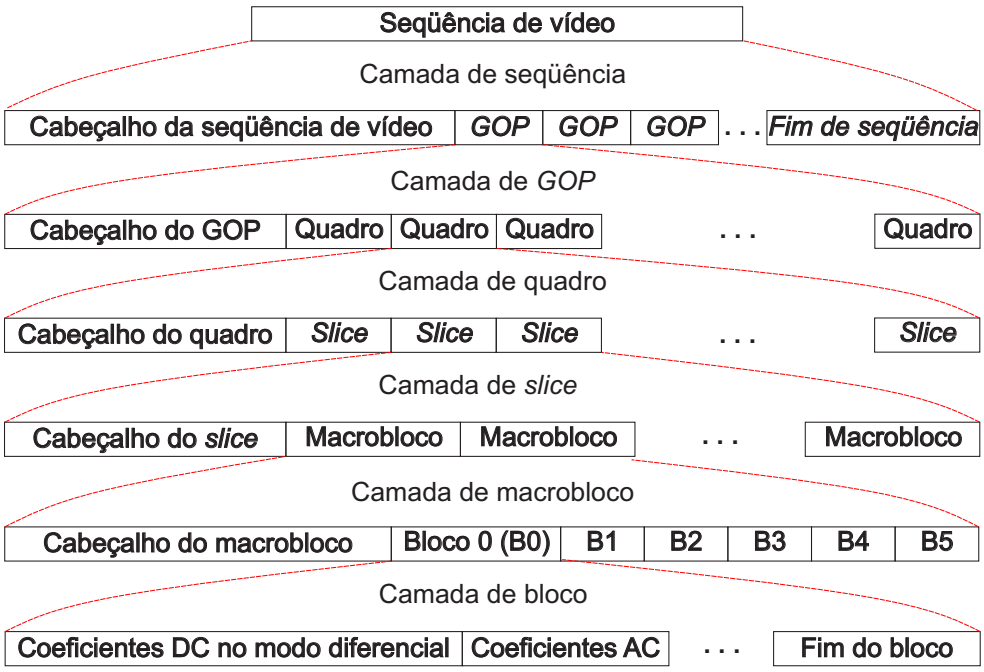


Figura 2.15: Camadas de um fluxo de bits do MPEG-2.

2.1.7 O padrão de codificação de vídeo H.264

O H.264/AVC (*Advanced Video Coder*) ou MPEG-4 Parte 10 [31] é o padrão de compressão de vídeo mais aceito no mercado e na academia desde a adoção do MPEG-2, desenvolvido em esforço conjunto pelo MPEG (*Motion Picture Experts Group*), que é um grupo de estudos pertencente à ISO (*International Standards Organization*) e o VCEG (*Video Coding Experts Group*), que é um grupo de estudos pertencente à ITU (*International Telecommunication Union*). A união destes grupos de estudos para padronização deste *codec* denomina-se JVT (do inglês, *joint video team*).

Este padrão permite taxas de compressão bem maiores do que já se conseguiu com os padrões anteriores, permitindo a compressão de vídeos com ou sem entrelaçamento de forma bastante eficiente e mesmo usando altas taxas de compressão oferecendo ainda uma qualidade visual melhor do que os padrões anteriores. Apesar disso, um grande esforço da indústria e da academia tem sido realizado para melhorar ainda mais o desempenho dos codificadores de vídeo. Uma nova chamada de propostas para compressão de vídeo denominado HEVC (*High Efficiency Video Coding*) iniciou em 2007 de forma que a padronização definitiva ocorra apenas em 2013 [48]. Um detalhe interessante que deve ser mencionado é que nos padrões de vídeo não se define um *codec* (codificador e decodificador) específico, mas apenas sintaxe de uma seqüência de bits de vídeo codificado juntamente com um método de decodificação dessa seqüência de bits.

A Figuras 2.16 mostra o diagrama de blocos básico do codificador do H.264. Na codificação o primeiro quadro deve ser necessariamente do tipo *intra*, equivalendo a uma codificação de imagem. Nele, o quadro é dividido em blocos de 16×16 (ou 8×8 pixels para crominância, no caso de utilizar o formato 4:2:0) e em cada bloco é feita uma predição desta informação a partir de pixels vizinhos previamente codificados. Entretanto para codificar um quadro explorando a redundância temporal, deve-se utilizar o modo *inter*, que utiliza a estimação de movimento para gerar uma predição baseada em um quadro previamente codificado. A diferença entre a informação original de um bloco e sua predição, denominado de resíduo, é transformado utilizando a DCT modificada [4, 5]. Em seguida, os coeficientes são quantizados e reordenados utilizando o ziguezague. A partir disso um codificador de entropia é aplicado tanto para os coeficientes dos resíduos como os símbolos que representem a predição, bits de controle, etc. Este processo pode ser repetido várias vezes em um bloco para se escolher o melhor modo de predição baseado no critério de minimização de uma função de custo $J = D + \lambda R$ que relaciona taxa e distorção, onde D se refere à distorção ou diferença (média ou quadrática) entre os blocos codificado e original, λ o multiplicador de Lagrange e

R a taxa (ou estimativa dela) de codificação. Independentemente do modo de codificação escolhido, o decodificador deve ser capaz de interpretar corretamente o sinal, caracterizando a otimização da relação taxa-distorção como um método ou sistema não-normativo. Em seguida, um filtro de redução de efeitos de blocos, em inglês *deblocking filter*, é aplicado. Após aplicar o processo de codificação em todo o quadro, sua reconstrução é guardada para que possa ser utilizada como referência para predições entre quadros. No processo de reconstrução, os coeficientes quantizados são re-escalados (operação "inversa" à quantização) e inversamente transformados, obtendo assim o resíduo, que deve ser somado à informação predita. É importante ressaltar que um certo nível de distorção foi introduzido, visto que a quantização é um processo não-reversível. Portanto, o bloco decodificado não é idêntico ao bloco original. Logo, o decodificador local dentro do codificador se faz necessário, pois tanto o codificador quanto o decodificador devem utilizar os mesmos quadros de referência para que seja feita a predição evitando assim um erro conhecido pela literatura como escorregamento, em inglês *drifting*. Na prática, este efeito geralmente deixa um rastro em torno dos objetos em movimento.

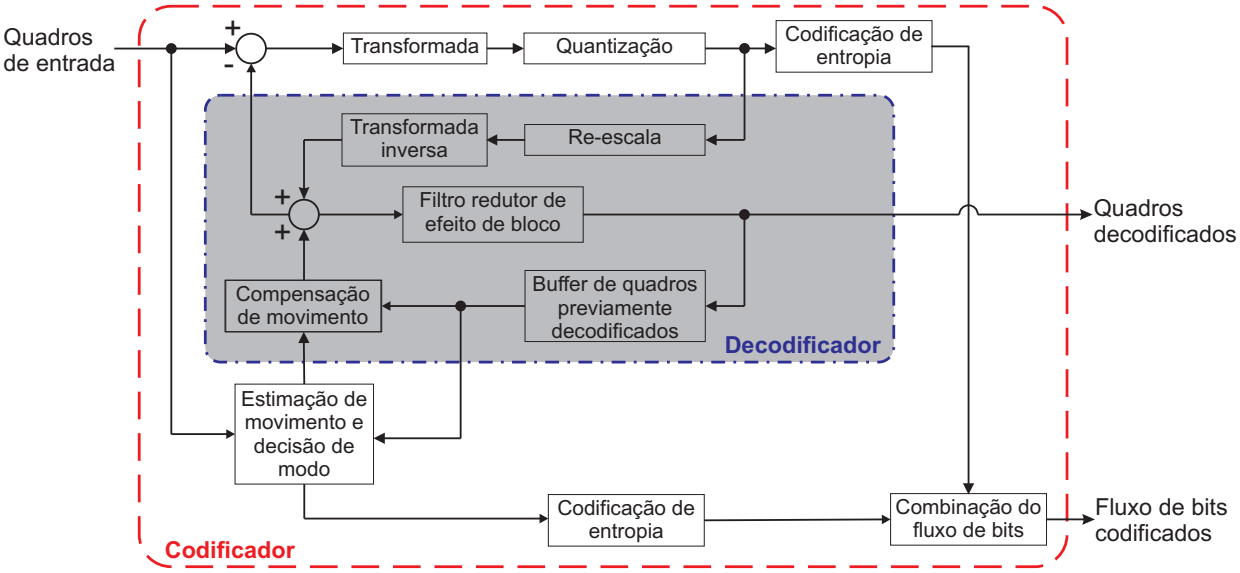


Figura 2.16: Diagrama de blocos do codificador de vídeo do H.264.

No codificador, os quadros de vídeo são processados de forma a serem reduzidos a um fluxo de bits. Já no decodificador, essa seqüência de bits comprimida é decodificada para que seja produzida uma versão reconstruída dos quadros de vídeo originais. O diagrama de blocos do decodificador H.264 é mostrado na Figura 2.17. Primeiramente, o fluxo de bits comprimido é submetido à decodificação de entropia, onde o interpretador (*parser*) distingue entre as informações de cabeçalho, modos de predição, informação residual, etc. As informações de predição, juntamente com a reconstrução dos resíduos, formam os quadros

decodificados. Os coeficientes dos blocos residuais estão sujeitos ao reordenamento (processo inverso ao ziguezague). Em seguida, os coeficientes de cada bloco quantizado são re-escalados e inversamente transformados. Finalmente, a somatória entre o bloco (*intra* ou *inter*) predito e o bloco residual formam um bloco decodificado. Por último, todos os macroblocos são ordenados e filtrados pelo filtro redutor de efeitos de bloco para que se produza a versão reconstruída do quadro.

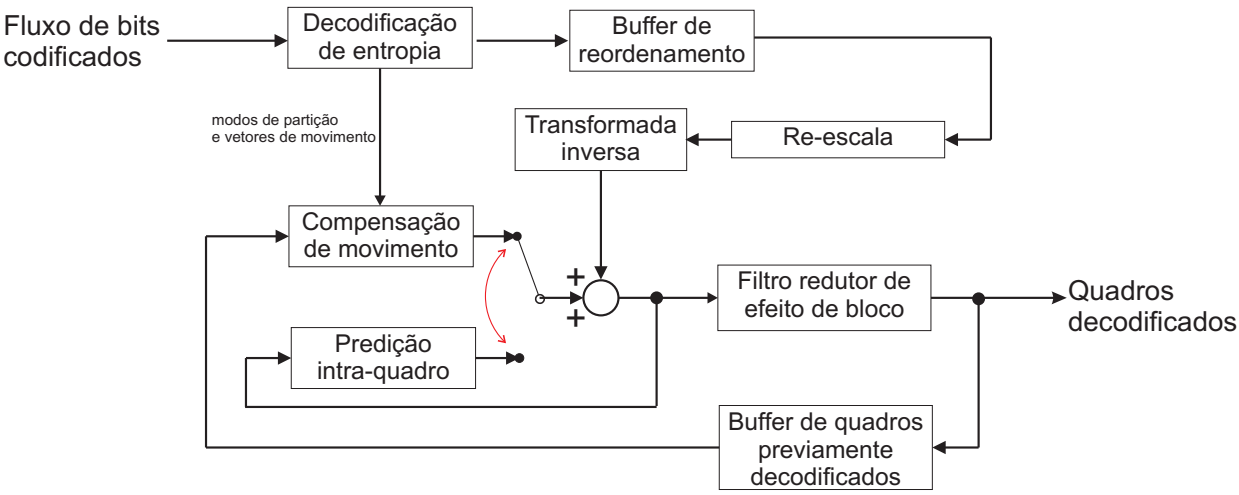


Figura 2.17: Diagrama de blocos do decodificador de vídeo do H.264.

2.1.7.1 Divisão de um quadro em macroblocos e *slices*

Todos os quadros são particionados em macroblocos de tamanho fixo de 16×16 amostras de componente de luminância. No caso de um vídeo colorido que utiliza o formato YUV 4:2:0, dois blocos de 8×8 amostras são utilizadas para as crominâncias. Todas as amostras (luminância e crominância) de um macrobloco são espacialmente ou temporalmente preditos, e o resíduo resultante (caso exista) é representado utilizando uma codificação por transformada. Os macroblocos são organizados em *slices*, que representam regiões lexicográficas de um dado quadro que podem ser decodificados entropicamente de maneira independentemente entre si. O H.264 suporta cinco tipo de *slices*. No mais simples: o *slice I* (onde *I* significa *intra*), todos os macroblocos contidos nele são codificados sem se referir a nenhum outro quadro da seqüência de vídeo. Quadros já comprimidos anteriormente podem ser utilizados para prever os macroblocos de *slices* do tipo *P* (preditivo) e *B* (bi-preditivo). Os outros dois tipos de *slices* são SP(*switching P*) e SI(*switching I*), que foram especificados para chavear eficientemente entre códigos com fluxo de bits comprimidos em várias taxas. [49]

2.1.7.2 Predição Espacial *Intra*

Conforme mencionado anteriormente, cada macrobloco pode ser transmitido como sendo uma das várias possibilidades de codificação dependendo do tipo de *slice*. Em todos os tipos de *slice*, pelo menos dois tipos de codificação de macroblocos *intra* são suportados, cuja predição é implementada no domínio espacial, sendo ainda distinguíveis apenas pelas suas dimensões das amostras de luminância: 4×4, 8×8 e 16×16. Já os pixels de crominância são preditos de maneira análoga, mas com tamanhos compatíveis ao de seu macrobloco. Neste tipo de predição os pixels vizinhos de blocos já codificados (eventualmente transmitidos) e decodificados são utilizados como referência para a predição.

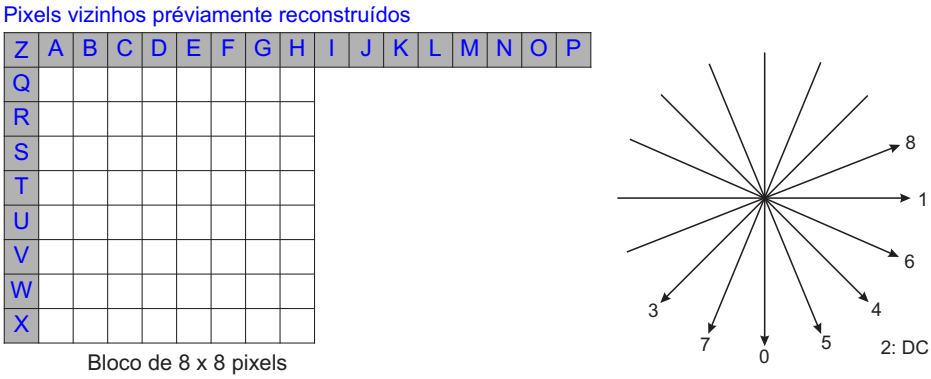


Figura 2.18: Predição *intra* quadro para um bloco de luminância de 8×8 pixels.

Na Figura 2.18 exemplifica-se a predição *intra* de um bloco de 8×8 pixels, que é calculada com base nas amostras A-P e Q-X, de acordo com as direções 0, 1, 3, 4, 5, 6, 7, 8. No caso do modo 2 (DC) todas as amostras são preditas com base em A-H e Q-X, como detalhado a seguir:

- *Modo 0*: Vertical

  As amostras A-H são extrapoladas verticalmente.

- *Modo 1*: Horizontal

  As amostras Q-X são extrapoladas horizontalmente.

- *Modo 2*: DC

  Todas as amostras são preditas a partir da média das amostras A-H e Q-X.

- *Modo 3*: Diagonal abaixo à esquerda

  As amostras são interpoladas em um ângulo de 45° a partir do canto superior direito (P).

- *Modo 4*: Diagonal abaixo à direita

  As amostras são interpoladas em um ângulo de 45° a partir do canto superior esquerdo (Z).

- *Modo 5*: Vertical direita)

  As amostras são interpoladas em um ângulo de 63,4° a partir do canto superior esquerdo (Z).

- *Modo 6*: Horizontal abaixo

  As amostras são interpoladas em um ângulo de 26,6° a partir do canto superior esquerdo (Z).

- *Modo 7*: Vertical esquerda

  As amostras são interpoladas em um ângulo de 63,4° a partir do canto superior direito (P).

- *Modo 8*: Horizontal acima

  As amostras são interpoladas em um ângulo de 26,6° a partir do canto inferior esquerdo (Z).

2.1.7.3 Predição *Inter*

Os macroblocos do tipo *inter* são preditos a partir da informação de quadros previamente codificados. No H.264 os macroblocos de tamanho 16×16 podem ser particionados, para melhor descrever o movimento dos objetos em cena, em duas regiões nas seguintes formas: 16×8 ou 8×16; ou em quatro sub-macroblocos de 8×8 pixels, que por conseguinte, podem ser particionados em regiões de 8×4, 4×8 ou 4×4 pixels. De acordo com a ilustração na Figura 2.19.

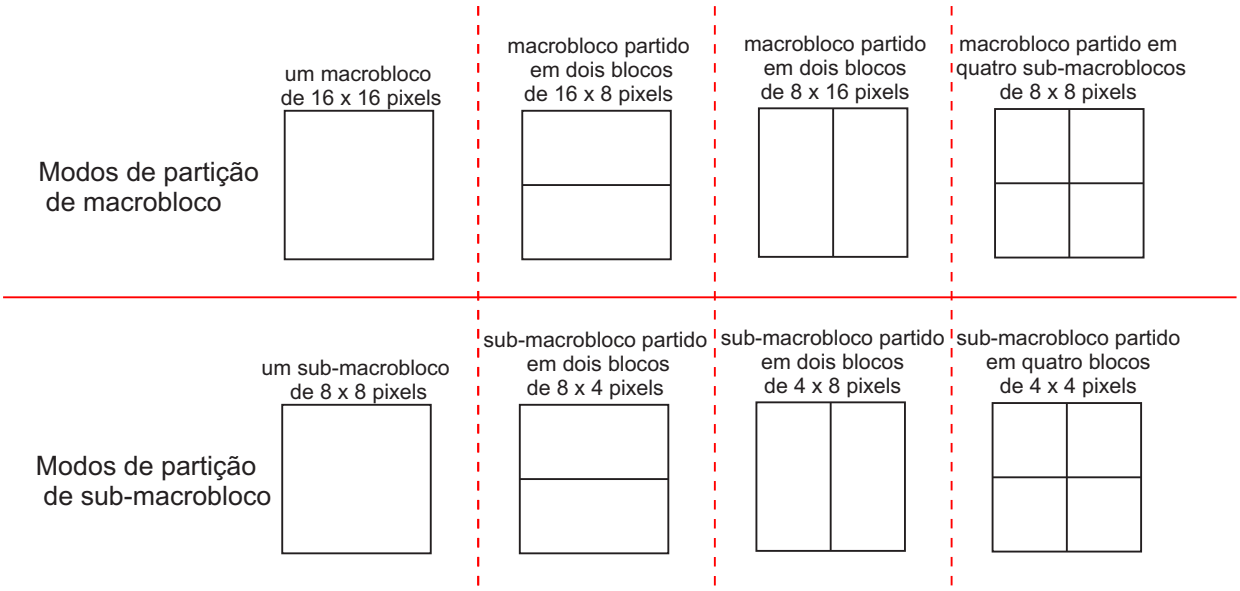


Figura 2.19: Tipos de macroblocos e sub-macroblocos.

O sinal predito de cada bloco de luminância é obtido por um deslocamento especificado por um vetor de movimento translacional e um índice que informa o quadro de referência. A precisão do vetor de movimento chega à granularidade de um quarto da distância entre pixels vizinhos, como mostra a Figura 2.20. Se o vetor de movimento aponta para uma posição inteira, a predição do sinal corresponde às amostras do quadro de referência. Caso contrário, a predição do sinal é obtida utilizando interpolação entre as posições inteiras. Os valores da predição em meio pixel é obtido aplicando um filtro FIR unidimensional de seis *taps*. Já os valores de predição com posições com valores referentes a um quarto de pixel são gerados pela média das amostras entre as posições inteiras e de meio pixel (observe a Figura 2.20). A predição dos valores para as componentes de crominância é obtida por interpolação bilinear.

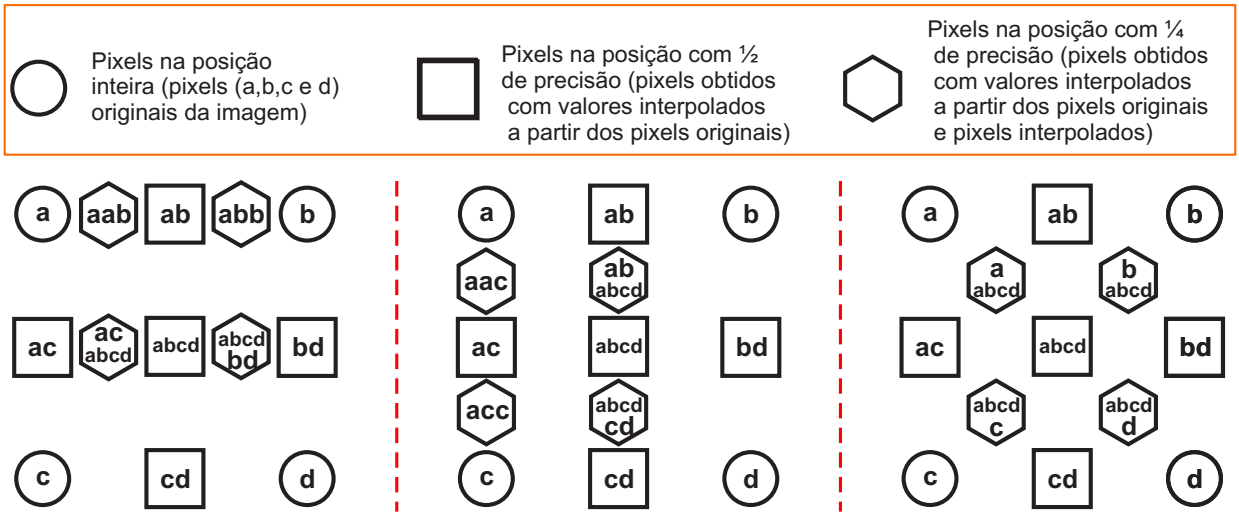


Figura 2.20: Ilustração das operações horizontal, vertical e diagonal em torno do pixel central '**abcd**' e ilustração de pixels nas precisões de um quarto, meio e inteiro.

Os vetores de movimento também são codificados diferencialmente, ao utilizar como predição a mediana dos vetores associados aos blocos vizinhos. Este processo é conhecido na literatura como *motion vector prediction*. Note que nenhuma predição das componentes dos vetores de movimento (ou outra forma de predição) pode ocorrer entre a fronteira dos *slices*, já que conceitualmente os *slices* devem ser decodificados independentemente.

O bloco bipredito é obtido pela média ponderada das predições de quaisquer pares de quadros de referência. Para tanto foram utilizadas duas listas de forma a indexar múltiplos quadros dos *buffers* denominados *list 0* e *list 1*, que são respectivamente referentes aos quadros anteriores e posteriores ao quadro atual.

Outra forma de predição *inter* é o *skip mode*. Para este modo, nenhum resíduo, vetor de movimento ou parâmetros de referência são transmitidos. A reconstrução do sinal é computada de maneira similar à predição de um macrobloco de tamanho 16×16 e quadro de referência mais próximo. Diferentemente dos padrões de vídeo antecessores, os vetores de movimento utilizados para reconstrução de um macrobloco do tipo *skip* é inferido de acordo com o movimento dos macroblocos vizinhos previamente decodificados ao invés de assumí-las como zero (ou seja, sem movimento) como no MPEG-2.

2.1.7.4 Transformada, Escalonamento e Quantização

Como mencionado anteriormente, o H.264 também utiliza uma codificação híbrida, onde existe uma etapa de predição seguida da codificação da informação residual por transformada. Contudo, em contraste com padrões anteriores, como o MPEG-2 ou H.263, que utilizam a transformada bidimensional de cossenos discreta (DCT-2D) de tamanho 8×8, o H.264 faz uso de um conjunto de transformadas inteiras de blocos de tamanhos diferentes. De modo geral, a transformada inteira de 4×4 é aplicada no resíduo da predição tanto para as componentes de luminância quanto para as de crominância. Além disso, a transformada Hadamard é aplicada para todos os coeficientes DC resultante de um macrobloco (16×16) que é codificado utilizando codificação *intra*. Apesar da importante aplicação em sistemas de baixa complexidade computacional, o uso de uma transformada de tamanho reduzido no H.264 tem ainda a vantagem de reduzir artefatos de *ringing* oriundos do fenômeno de Gibbs [50, 51]. Todavia, para vídeo de alta fidelidade, a preservação da suavidade e da textura é geralmente beneficiada com representações com funções de bases maiores. Um bom custo benefício para esta situação ocorre com o uso da transformada de tamanho 8×8. Uma transformada inteira parecida com a DCT bidimensional de tamanho 8×8 foi incorporada ao FRExt (vide a Seção 2.1.7.7), possibilitando implementações eficientes em sistemas com aritmética inteira. De fato, qualquer transformada inteira do H.264, assim como suas respectivas transformadas inversas, podem ser implementadas de maneira simples e eficiente, já que apenas as operações de deslocamento e adição em um processamento com $(8 + b)$ bits são necessários para comprimir e descomprimir um vídeo com b bits de profundidade.

Para quantizar os coeficientes transformados do H.264 utiliza-se um dos 52 possíveis valores de escalonamento dos quantizadores de reconstrução uniforme (URQs - *Uniform-Reconstruction Quantizers*), denominados de parâmetros de quantização ou simplesmente QP (do inglês, *Quantization Paremeters*). A escala de operação é organizada de forma que o passo de quantização dobra a cada incremento de seis

no valor de QP. Os coeficientes transformados são quantizados como na Equação 2.5 e em seguida são percorridos via ziguezague e processados por um codificador de entropia, que será descrito a seguir.

2.1.7.5 Codificação de Entropia

No H.264 vários elementos sintáticos são codificados utilizando a mesma estrutura de código de tamanho variável (VLC - *Variable Length Code*) denominado código exponencial-Golomb de ordem zero [4]. Alguns elementos sintáticos são codificados usando representação de códigos em tamanho fixo. Para os demais elementos sintáticos, duas possibilidades de codificação de entropia podem ser utilizadas. Quando utiliza-se a primeira configuração de codificação de entropia, que requer implementações de baixa complexidade computacional, o código exponencial-Golomb [4] é usado em quase todos os elementos sintáticos exceto para os coeficientes transformados e quantizados, que utiliza um método um pouco mais sofisticado denominado CAVLC - *context-adaptive variable length coding*. Quando o CAVLC é utilizado, o codificador chaveia entre diferentes tabelas de códigos de tamanho variável dependendo dos valores previamente decodificados, adicionando assim uma característica de contexto adaptativo. As tabelas de VLC foram desenvolvidas de forma que o contexto se comporte como uma probabilidade condicional. O desempenho do codificador de entropia aumenta sensivelmente ao utilizar a segunda configuração, referida na literatura como CABAC - *Context-Based Adaptive Binary Arithmetic Coding* [35, 52]. Comparado ao CALVC, o CABAC tipicamente reduz em torno de 10 a 20% a taxa de bits para a mesma qualidade objetiva de um sinal de vídeo SDTV/HDTV codificado [35].

2.1.7.6 Filtro Redutor de Efeito de Bloco

Uma das características particulares dos codificadores baseados em blocos é a ocorrência de descontinuidades visualmente perceptíveis ao longo das bordas dos blocos, uma vez que as predições são realizadas em blocos e os resíduos são independentemente codificados. Por esta razão, o H.264 define um filtro de redução de efeitos de bloco adaptativo (igual ao do H.263) aplicável ao ciclo de codificação e reconstrução, e isso se constitui como uma componente necessária para o processo de decodificação. A adaptabilidade do filtro ocorre desde o nível de *slices*, passando pelas bordas até o nível de amostras. Os parâmetros do filtro são controlados pelos valores de vários elementos sintáticos. Para maiores detalhes vide a referência [28, 53]. Como resultado, o efeito de bloco é reduzido sem afetar muito as altas freqüências do conteúdo da imagem. Conseqüentemente, a qualidade subjetiva aumenta significativamente, ao mesmo

tempo em que o filtro reduz tipicamente entre 5 a 10 % de taxa de bits produzindo a mesma qualidade objetiva em comparação com um vídeo não filtrado [53]. Além disso, permite que as predições sejam feitas com maior eficiência, pois o efeito de bloco pode viciar uma predição.

2.1.7.7 Perfis do H.264

Os perfis e níveis especificam pontos de conformidade que permitem a interoperabilidade entre várias aplicações que tenham requisitos funcionais similares. Um perfil define um conjunto de ferramentas de codificação ou algoritmos que podem ser usados para gerar um fluxo de bits, já os níveis indicam restrições em certos parâmetros chave do fluxo de bits. Todos os decodificadores de um determinado perfil devem ter a capacidade de suportar todas as características deste mesmo perfil. Já os codificadores não possuem a obrigação de utilizar nenhuma característica específica de um perfil, bastando gerar um fluxo de bits compatível, ou seja, que algum decodificador H.264 em conformidade com o padrão e o perfil desejado consiga decodificar.

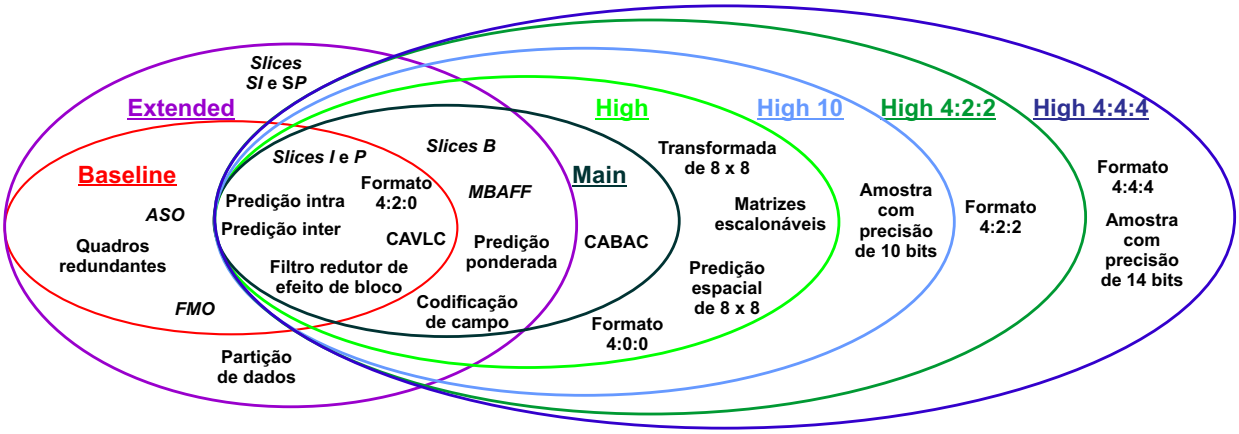


Figura 2.21: Ilustração dos perfis do H.264.

Na primeira versão do H.264 três perfis foram definidos: *Baseline*, *Extended* e *Main*, onde o perfil *Baseline* suporta todas as características do H.264 versão 1 (2003), exceto pelos conjuntos de características a seguir:

1. *Slices B*, codificação de campo, chaveamento adaptativo entre quadro e campo (MBAFF - *Macroblock Adaptive Switching Between Frame and Field*) e predição ponderada.
2. CABAC.
3. *Slices SI* e *SP*, e partição de dados com *slice*.

Os dois primeiros itens contêm um conjunto de características que são suportados pelo perfil *Main*, além das características suportadas pelo *Baseline* exceto para a FMO (*Flexible Macroblock Order*) e outras características de robustez a erros. [49]. O perfil *Extended* suporta todas as características do perfil *Baseline* adicionado aos itens um e três. A grosso modo, o perfil *Baseline* foi desenvolvido visando aplicações com o mínimo de complexidade computacional e o máximo de robustez a erro, já o perfil *Main* focava aplicações que necessitassem do máximo em eficiência de compressão. Finalmente, o perfil *Extended* foi desenvolvido para promover um compromisso entre os perfis *Baseline* e *Main* com um foco para necessidades específicas de aplicações com *streaming* de vídeo adicionado à robustez a erros e perda de pacotes.

A Figura 2.21 mostra os perfis *High* (*High*, *High 10*, *High 4:2:2* e *High 4:4:4*). Este conjunto de perfis é conhecido na literatura como FREext (*Fidelity Range Extension*), que adiciona aos perfis anteriores a possibilidade de se utilizar transformadas de tamanho 8×8 amostras e predição espacial em blocos de tamanho 8×8 pixels. O perfil *High* faz uso, assim como no perfil *Main*, de uma precisão de 8 bits por amostra para seqüências no formato 4:2:0, em aplicações típicas de SD e HD. Outros dois perfis chamados *High 10* e *High 4:2:2* extendem a capacidade do padrão de incluir demandas que necessitem de amostras com maior precisão (maior que 10 bits por amostra) e maior formato de crominância (no caso, 4:2:2). Finalmente, o FREext ainda possui a especificação do perfil *High 4:4:4*, que além de não sub-amostrar a crominância, uma precisão de 14 bits por amostra pode ser utilizada.

2.1.8 Métrica de qualidade de imagens e vídeos

No processo de compressão de um vídeo digital, geralmente se aceita perdas entre o vídeo comprimido e o vídeo original, em detrimento de obter maiores taxas de compressão. Neste ponto, faz-se necessária a questão: até que ponto pode-se aceitar essa perda na qualidade entre o vídeo comprimido e o vídeo original? Como medí-la? Em sistemas de comunicação a qualidade do sinal recebido é medida pela razão entre a potência do sinal e a potência do ruído, ou relação sinal-ruído (SNR). Embora a SNR não seja perfeita, ela permite definir claramente se um determinado sinal recebido é melhor ou pior do que outro, além de permitir com precisão qual a taxa de erro esperada em cada sinal, e até mesmo se a demodulação é possível. A SNR é uma medida objetiva, isto é, ela pode ser medida com precisão e não muda com a repetição do experimento, é amplamente utilizada em sistemas de comunicação, via rádio, cabo ou qualquer outro.

Na área de processamento de imagens e vídeo digitais utiliza-se a PSNR, que apesar de ser uma medida objetiva e de ser a métrica de qualidade mais utilizada, tem os seus opositores [54,55]. Os argumentos mais

comuns são: ela requer a imagem original para cálculo, o que nem sempre é possível, e às vezes contradiz a percepção subjetiva de qualidade. Apesar disso, em casos típicos, quanto maior a PSNR melhor é a percepção subjetiva da imagem, ou seja, a métrica objetiva da PSNR é correlacionada com a métrica subjetiva. A percepção subjetiva de qualidade se refere à como uma pessoa percebe a diferença entre as imagens. Isso pode depender muito do que a pessoa fará da imagem, com a atenção que ela dá à imagem, com as experiências pessoais do observador, e muitas outras coisas, fazendo com que os resultados de uma medida subjetiva não possam ser repetidos nem aferidos com precisão.

A maior diferença de resultados entre a percepção subjetiva e a PSNR é que esta dá pesos iguais a todos os pixels da imagem. Uma grande variação em um único pixel (ou região), em geral, não degrada muito a PSNR de uma imagem, enquanto o sistema visual humano é capaz de perceber essas variações. Exemplos comuns nesse tópico: se o valor de todos os pixels de uma imagem for subtraído por um valor unitário ou que tenham todos os pixels deslocados, a PSNR degrada muito, e um observador humano poderá afirmar que a imagem não foi deteriorada; no caso em que as áreas da imagem em que o observador geralmente foca ou observa com maior atenção possuírem alta qualidade, e as áreas de fundo (onde o observador normalmente não foca) estiverem degradadas, a PSNR vai piorar, mas o observador humano não vai mudar sua percepção. Esta característica visual é denominada de foveação [56].

Apesar dos problemas encontrados na medida, a PSNR é ainda a medida mais utilizada de qualidade de imagem e vídeo encontrada, por ser um teste objetivo bastante difundido e de fácil medição, desde que se tenha a imagem (ou seqüência de imagens) original e a reconstruída. Nesta tese os resultados principais utilizam a PSNR, pois permite uma comparação objetiva e não-viciada entre trabalhos similares ou do mesmo tópico.

2.2 REDIMENSIONAMENTO DE IMAGENS

O redimensionamento de imagem é um método de ampliação ou redução de uma imagem. Nesta tese, serão utilizadas alguma técnicas de redução, para diminuir a resolução de alguns quadros de um vídeo, de forma a gerar seqüências com resolução mista (vide as Seções 3.3 e 4). Já a ampliação de imagens, geralmente denominado de interpolação, é utilizada nos quadros de resolução menor antes do processo de realce apresentado nesta tese.

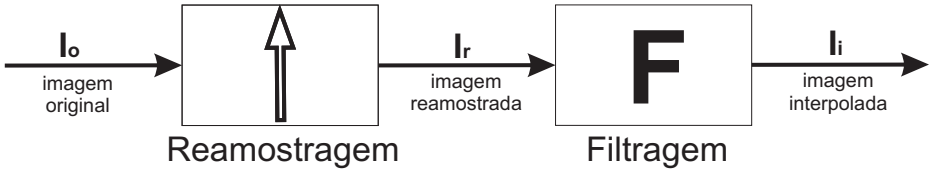


Figura 2.22: Processo de interpolação de uma imagem. Onde a imagem original é submetida ao processo de reamostragem (mostrada na Equação 2.12) seguida de uma filtragem.

Para se ampliar uma imagem $\mathbf{I}_o$ de dimensões $w \times h$ pixels por um fator inteiro s em casa direção, deve-se primeiramente reamostrar $\mathbf{I}_o$ para $\mathbf{I}_r$ da seguinte forma:

$$I_r(x,y) = \begin{cases} I_o\left(\frac{x}{s}, \frac{y}{s}\right) & : \ x,y \in \mathbb{Z}^+ \\ 0 & : \ caso \ \ contrário \end{cases} \tag{2.12}$$

onde $I_r(x,y)$ descreve um elemento (pixel) de $\mathbf{I}_r$ na posição (x,y). Assim, uma imagem reamostrada $\mathbf{I}_r$ de dimensões $ws \times hs$ são obtidos, mas com uma série de pixels 'faltantes' que poderiam se substituídos pelos valores de pixels vizinhos. Entretanto, este processo de interpolação pode gerar uma série de artefatos como, por exemplo, criação de bordas serrilhadas e baixo detalhamento das bordas. Devido a estes artefatos os algoritmos interpoladores geralmente filtram a imagem reamostrada $\mathbf{I}_r$, como mostra a Figura 2.22.

Os filtros (*kernels*) de interpolação $F(t)$ apresentados a seguir são descritos apenas em uma dimensão. Para estender estes filtros em duas dimensões, deve se fazer o produto de dois filtros ortogonais de uma dimensão nas direções t_x e t_y, ou seja:

$$F(t_x, t_y) = F(t_x)F(t_y). \tag{2.13}$$

Em seguida o filtro deve ser amostrado, resultando em $\mathbf{F}$ e convoluído[2] com a imagem reamostrada $\mathbf{I}_r$ para gerar a imagem ampliada ou interpolada:

$$\mathbf{I}_i = \mathbf{I}_r * \mathbf{F}. \tag{2.14}$$

Nesta tese, iremos utilizar a interpolação DCT, que será apresentada no Capítulo 4 com uma abordagem diferente. Além de outros três tipos de interpolações:

[2]A convolução para funções no domínio discreto é definida por: $I * F(n) = \sum_m I(m)F(n-m)$.

1. Bilinear: A interpolação bilinear é a aplicação do método linear no espaço bidimensional. O filtro $F(t)$ ou *kernel* utilizado para interpolação com este método é dado por [57]:

$$F(t) = (1 - |t|)^{+} \tag{2.15}$$

onde $(\cdot)^{+}$ denota a parte positiva e t representa a posição das amostras. Entretanto uma forma direta de descrever a interpolação linear é descrita matematicamente como [57]:

$$\begin{aligned} I_i(x,y) \;=\; & I_o\left(\left\lfloor \frac{x}{s} \right\rfloor s, \left\lfloor \frac{y}{s} \right\rfloor s\right)\left(1 - \left(x - \left\lfloor \frac{x}{s} \right\rfloor s\right)\right)\left(1 - \left(y - \left\lfloor \frac{y}{s} \right\rfloor s\right)\right) + \\ & I_o\left(\left\lceil \frac{x}{s} \right\rceil s, \left\lfloor \frac{y}{s} \right\rfloor s\right)\left(x - \left\lfloor \frac{x}{s} \right\rfloor s\right)\left(1 - \left(y - \left\lfloor \frac{y}{s} \right\rfloor s\right)\right) + \\ & I_o\left(\left\lfloor \frac{x}{s} \right\rfloor s, \left\lceil \frac{y}{s} \right\rceil s\right)\left(1 - \left(x - \left\lfloor \frac{x}{s} \right\rfloor s\right)\right)\left(y - \left\lfloor \frac{y}{s} \right\rfloor s\right) + \\ & I_o\left(\left\lceil \frac{x}{s} \right\rceil s, \left\lceil \frac{y}{s} \right\rceil s\right)\left(x - \left\lfloor \frac{x}{s} \right\rfloor s\right)\left(y - \left\lfloor \frac{y}{s} \right\rfloor s\right) \end{aligned} \tag{2.16}$$

onde os pixels da imagem interpolada $I_i(x,y)$ é obtido pela combinação linear das distâncias entre os pixels vizinhos conhecidos (valores da imagem original $\mathbf{I}_o$), a operação $\lfloor\cdot\rfloor$ representa o truncamento para o maior valor inteiro menor que o argumento e a operação $\lceil\cdot\rceil$ representa o arredondamento, que retorna o menor valor inteiro maior que o argumento.

2. Bicúbico: A interpolação bicúbica utiliza o seguinte filtro [57, 58]:

$$F(t) = \begin{cases} (a+2)\,|t|^3 - (a+3)\,|t|^2 + 1 & : \; se \;\; |t| \leq 1, \\ a\,|t|^3 - 5a\,|t|^2 + 8a\,|t| - 4a & : \; se \;\; 1 < |t| < 2, \\ 0 & : \; caso \;\; contrário \end{cases} \tag{2.17}$$

onde a é um parâmetro livre. Esta função é resultado de uma série de condições impostas pelo filtro interpolador. O filtro em questão é composto de uma função polinomial de terceira ordem definida por partes que possui as seguintes características: simétrico, contínuo e possuir a derivada primeira contínua. Essas condições geram apenas um grau de liberdade em a, que é geralmente igual a -1, $-0,75$ ou $-0,5$, motivados por várias noções de optimalidade [57].

3. Lanczos: Tanto na interpolação quanto na redução da imagem o filtro ideal seria uma função $sinc$, entretanto esta função tem comprimento infinito o que inviabiliza sua implementação. Portanto, o filtro ideal deverá ser limitado à uma janela implicando numa solução que gera vários artefatos. O filtro Lanczos é uma função $sinc$ janelada por uma janela $sinc$, sendo descrita matematicamente

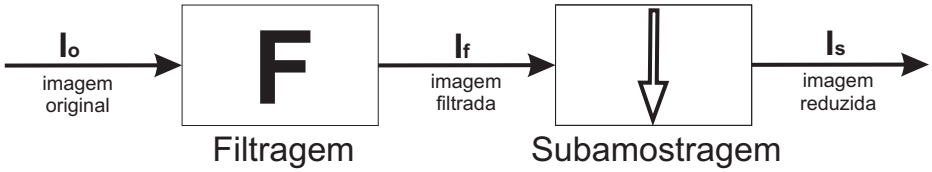


Figura 2.23: Processo de redução de uma imagem. Onde a imagem original é submetida ao processo filtragem seguido da reamostragem (mostrada na Equação 2.19).

por [57, 59]:

$$F(t) = \begin{cases} sinc(t)sinc(t/a) & : \ -a < t < a, \\ 0 & : \ caso \ contrário \end{cases} \tag{2.18}$$

onde a é um inteiro positivo, geralmente igual a 2 ou 3, que controla o tamanho do *kernel*.

A redução de imagem é realizada de maneira similar à ampliação, entretanto a etapa de filtragem ocorre antes do processo de reamostragem, como mostra a Figura 2.23.

A subamostragem é obtida pela exclusão linha-coluna por um fator s em cada direção:

$$I_s(x, y) = I_o(xs, ys) \ : \ x, y \in \mathbb{Z}^+ \tag{2.19}$$

para reduzir o *aliasing*[3] deve-se convoluir a imagem $\mathbf{I}_o$ com $\mathbf{F}$ (que pode ser o filtro bilinear, bicubico ou Lanczos, das Equações 2.15, 2.17, 2.18 amostrados e aplicados nas duas dimensões). Matematicamente podemos escrever:

$$\mathbf{I}_f = \mathbf{I}_o * \mathbf{F}, \tag{2.20}$$

que em seguida deve ser feita a subamostragem da imagem:

$$I_s(x, y) = I_f(xs, ys) \ : \ x, y \in \mathbb{Z}^+. \tag{2.21}$$

Note que para ampliar ou reduzir por fatores não inteiros, pode-ser realizar uma interpolação seguida de uma redução para o tamanho especificado.

[3]O *aliasing* é uma sobreposição de espectros no domínio da transformada de Fourier e ocorre quando a freqüência de amostragem é menor que a freqüência de Nyquist, ou seja, menor que duas vezes a freqüência máxima do sinal.

2.3 SUPER-RESOLUÇÃO

A super-resolução (SR) é uma técnica utilizada para se obter uma imagem com resolução maior que a obtida pelo dispositivo de aquisição [60, 61], cujo objetivo é recuperar detalhes de uma imagem pela utilização da informação contida em um conjunto de imagens, realçando características importantes como a borda dos objetos. Tradicionalmente, a super-resolução de uma imagem é obtida por meio da utilização de um conjunto de imagens de baixa-resolução que possuam alta correlação, onde são exploradas pequenas variações de informação presentes nas imagens de baixa-resolução para gerar uma nova imagem de alta resolução com maior detalhamento. O exemplo mais comum de detalhamento das imagens de baixa-resolução é o movimento sub-pixel, onde o deslocamento entre as imagens é fracionário com relação às posições dos pixels nos quadros de baixa-resolução. Entretanto, as pesquisas em torno da super-resolução podem ser classificadas da seguinte forma:

1. super-resolução baseada em restauração, onde a formação da imagem de alta resolução é baseada em várias imagens em baixa-resolução;

2. super-resolução baseada em aprendizagem, cuja formação da imagem de alta resolução é baseado em inferências a partir da análise de várias imagens em alta resolução.

2.3.1 Modelos utilizados na super-resolução

2.3.1.1 Modelos de aquisição linear de imagem

O modelo de aquisição de imagens descreve o processo de sensoriamento de uma cena. Neste modelo, os dados observados são imagens de baixa-resolução, cujo modelo em [62] pode ser descrito como:

$$\mathbf{g}_k = \mathbf{D}_k \mathbf{f}_k + \boldsymbol{\eta}_k \tag{2.22}$$

onde $\mathbf{g}_k$ é um vetor de tamanho $N \times 1$, que representa a imagem em baixa-resolução no instante k. Os elementos de $\mathbf{g}_k$ correspondem aos pixels da imagem de tamanho $N_w \times N_h$ ordenados lexicograficamente. O vetor $\mathbf{f}_k$ de tamanho $M \times 1$, representa a imagem de alta resolução, onde $M = M_w M_h$, representando uma imagem de tamanho $M_w \times M_h$. Define-se a redução do tamanho da imagem na direção horizontal e vertical como: $R_w = M_w/N_w$ e $R_h = M_h/N_h$, respectivamente. O vetor $\boldsymbol{\eta}_k$ representa o ruído de aquisição, assumido como estatisticamente independente no tempo. A matriz $\mathbf{D}_k$ representa o sistema de

aquisição de imagem discretizado, de tamanho $N \times M$ no instante k. A coluna m de $\mathbf{D}_k$ representa a função de espalhamento dos pixels em m da imagem de alta-resolução durante a aquisição. Já a linha n de $\mathbf{D}_k$ representa a função de aquisição dos pixels em n da imagem em baixa-resolução, cujos elementos são os coeficientes de uma combinação linear entre os pixels da imagem de alta resolução para a formação dos pixels em baixa-resolução. Em outras palavras, $\mathbf{D}_k$ representa as distorções inerentes do processo de aquisição da imagem. Essas distorções podem se originar na óptica, que são causadas por lentes e fotosensores, e também na subamostragem, que é a redução no número de amostras obtidas pelo sensor. Em alguns casos específicos, onde $R = R_w = R_h$ e é inteiro, e as distorções causadas na lente e no sensor são modeladas como operadores invariantes no espaço, a matriz $\mathbf{D}_k$ pode ser separada, podendo reescrever a Equação 2.22 para

$$\mathbf{g}_k = \mathbf{S}_k \mathbf{B}_k \mathbf{f}_k + \boldsymbol{\eta}_k \tag{2.23}$$

onde $\mathbf{S}_k$ é a matriz $N \times M$ de subamostragem, que representa a redução da resolução pela dizimação das amostras da imagem. A matriz $\mathbf{B}_k$ é uma matriz $M \times M$ que modela a convolução bidimensional da distorção óptica com a imagem.

2.3.1.2 Modelo combinado de aquisição e movimento de imagens

O modelo de movimento da imagem parte do pressuposto de que uma imagem no instante j pode ser composta pela imagem no instante k, com movimento compensado e adicionada a uma informação residual que não pode ser obtida da imagem no instante k. Isto pode ser matematicamente descrito por:

$$\mathbf{f}_j = \mathbf{M}_{j,k} \mathbf{f}_k + \mathbf{e}_{j,k} \tag{2.24}$$

onde $\mathbf{f}_j$ e $\mathbf{f}_k$ são vetores que representam as imagens $\mathbf{f}$ em instantes de tempo diferentes. A matriz $\mathbf{M}_{j,k}$ representa a transformação de movimento dos conteúdos no instante k para o instante j. Já a matriz $\mathbf{e}_{j,k}$ é o erro de movimento (ou informação residual), que representa uma nova informação, no instante j, que não pode ser obtida após aplicar a transformação de movimento na imagem no instante k.

Ao combinar os modelos de aquisição descrito pela Equação 2.23, com o modelo de movimento dado pela Equação 2.24, obtemos o seguinte modelo:

$$\begin{aligned}\mathbf{g}_j &= \mathbf{D}_j\left(\mathbf{M}_{j,k}\mathbf{f}_k + \mathbf{e}_{j,k}\right) + \boldsymbol{\eta}_k \\ &= \mathbf{D}_j\mathbf{M}_{j,k}\mathbf{f}_k + \mathbf{D}_j\mathbf{e}_{j,k} + \boldsymbol{\eta}_k \\ &= \mathbf{C}_{j,k}\mathbf{f}_k + \varepsilon_{j,k} \end{aligned} \tag{2.25}$$

onde $\mathbf{C}_{j,k}$ é a matriz de transformação conjunta de movimento e aquisição. O erro $\varepsilon_{j,k}$ é a soma dos erros de aquisição e de movimento. A Equação 2.25 ser representada em uma forma mais compacta por:

$$\boldsymbol{g} = \mathbf{C}_k\mathbf{f}_k + \boldsymbol{\varepsilon}_k \tag{2.26}$$

onde $\boldsymbol{g} = \left[\mathbf{g}_1^T \cdots \mathbf{g}_L^T\right]^T$, $\boldsymbol{\varepsilon}_k = \left[\varepsilon_{1,k}^T \cdots \varepsilon_{L,k}^T\right]^T$ e $\mathbf{C}_k$ é uma matriz de tamanho $LN \times M$, assumindo que o conjunto de imagens capturadas é formado por L imagens.

Apesar de similares ao modelo anteriormente apresentado na Equação 2.22, este é mais complexo, pois apresenta dificuldades na modelagem estatística de $\varepsilon_{j,k}$ devido à presença de *outliers*[4]. Para um bom funcionamento do modelo, os *outliers* são detectados e excluídos do modelo [63, 64].

2.3.1.3 Modelos de imagem

O problema da super-resolução, é no geral, mal posto, seja porque não possua solução, possua infinitas soluções ou porque a solução é muito sensível ao ruído. Para se resolver tal problema e se obter uma solução única e estável, os algoritmos utilizam algumas informações adicionais sobre a imagem. Nos algoritmos Bayesianos, a informação adicional é definida na forma de uma distribuição *a priori* por meio de treinamentos [65], enquanto que nos algoritmos determinísticos a informação adicional é tratada como restrição ou como penalidade de regularização [62, 66]. A informação adicional mais comumente assumida é de que as variações de intensidade de uma imagem $\mathbf{f}_k$ são relativamente pequenas [67] e são geralmente modeladas da seguinte forma:

$$\|\mathbf{R}_k\mathbf{f}_k\| = \delta_{\mathbf{f}_k} \tag{2.27}$$

[4]O termo *outlier* foi inicialmente utilizado na estatística. Na literatura um *outlier* é um elemento ou uma medida ruim, geralmente um erro de grande magnitude, que não segue o modelo assumido

onde $\|\cdot\|$ é a operação de cálculo da norma ℓ_n, ou outra medida de distância escolhida, e $\mathbf{R}_k$ é uma matriz de tamanho $P \times N$ que contém os coeficientes de regularização. Em outras palavras, $\mathbf{R}_k$ representa uma operação "passa-altas", portanto o vetor resultante de $\mathbf{R}_k\mathbf{f}_k$ representa as variações de intensidade de $\mathbf{f}_k$ e $\delta_{\mathbf{f}_k}$ é uma medida conhecida de $\|\mathbf{R}_k\mathbf{f}_k\|$.

Um modelo muito utilizado para definir $\mathbf{f}_k$ é o campo aleatório de Markov (MRF - *Markov Random Field*) [68], que é especificado por meio de uma distribuição de Gibbs [66, 67] com a seguinte densidade de probabilidade:

$$\rho(\mathbf{f}_k) = \frac{1}{q} e^{-Q(\mathbf{f}_k)/\beta} \tag{2.28}$$

onde q é a constante para normalização da distribuição, β o parâmetro de controle relacionado ao desvio-padrão da distribuição e $Q(\mathbf{f}_k)$ pode ser definido como:

$$Q(\mathbf{f}_k) = \sum_i V\left([\mathbf{R}_k\mathbf{f}_k]_i\right), \tag{2.29}$$

onde $[\mathbf{R}_k\mathbf{f}_k]_i$ é um elemento do resultado de $\mathbf{R}_k\mathbf{f}_k$ e $V\left([\mathbf{R}_k\mathbf{f}_k]_i\right)$ é uma função que aplica um potencial a este elemento. As funções de potencial mais encontradas na literatura são: quadrática, valor absoluto, Huber [69], valor absoluto elevado à potência [70], dentre outros [62, 66, 71]. O modelo de imagem, com especificação $\mathbf{R}_k$, do conjunto de β e do potencial $V(\cdot)$, descreve a informação adicional necessária na busca de imagem de alta resolução [72].

2.3.2 Métodos de super-resolução

Nesta seção serão apresentados alguns algoritmos de super-resolução.

2.3.2.1 Interpolação não-uniforme e restauração

Neste método o problema de super-resolução é separado em duas etapas: a interpolação e a restauração. Na primeira etapa, uma imagem $\mathbf{h}_k$, com as dimensões da alta resolução, é criada a partir das imagens de baixa-resolução. Entretanto, esta imagem ainda possui distorções ópticas, que serão corrigidas pela restauração. Este método se baseia no modelo combinado (Seção 2.3.1.2), restrito para sistemas de

aquisição invariantes e a utilização do modelo de distorção óptica, permitindo a seguinte igualdade: $\mathbf{C}_{j,k} = \mathbf{S}_j\mathbf{M}_{j,k}\mathbf{B}_k$. Portanto, pode-se reformular a Equação 2.25, obtendo o seguinte modelo [72, 73]:

$$\begin{aligned} \mathbf{g}_j &= \mathbf{C}_{j,k}\mathbf{f}_k + \varepsilon_{j,k} \\ &= \mathbf{S}_j\mathbf{M}_{j,k}\mathbf{B}_k\mathbf{f}_k + \varepsilon_{j,k}. \end{aligned} \tag{2.30}$$

Neste caso, a imagem de baixa-resolução $\mathbf{g}_j$ é modelada pela imagem de alta-resolução $\mathbf{f}_k$ multiplicada por: uma distorção óptica $\mathbf{B}_k$, uma matriz $\mathbf{M}_{j,k}$ que representa a transformação de movimento dos conteúdos no instante k para o instante j e uma matriz de subamostragem $\mathbf{S}_j$. Já a matriz $\varepsilon_{j,k}$ é o erro ou a informação que não pode ser obtida após a aplicação das matrizes que representam o modelo de aquisição.

Em seguida, a solução da Equação 2.30 é separada em interpolação não-uniforme e restauração. A etapa de interpolação utiliza as imagens em baixa-resolução pelo seguinte modelo:

$$\mathbf{g}_j = \mathbf{S}_j\mathbf{M}_{j,k}\mathbf{h}_k + \varepsilon_{j,k}, \tag{2.31}$$

onde $\mathbf{h}_k$ é a imagem interpolada, que pode ser estimada utilizando uma interpolação não-uniforme:

$$\widehat{\mathbf{h}}_k = \mathbf{E}_k\boldsymbol{g} \tag{2.32}$$

onde $\widehat{\mathbf{h}}_k$ é uma estimativa de $\mathbf{h}_k$, $\boldsymbol{g} = \left[\mathbf{g}_1^T \cdots \mathbf{g}_L^T\right]^T$ é o conjunto de imagens de baixa-resolução e $\mathbf{E}_k$ a matriz de interpolação. Em [74–76] podem ser encontrados alguns métodos para encontrar a matriz $\mathbf{E}_k$. A imagem interpolada $\mathbf{h}_k$ pode ser restaurada utilizando:

$$\widehat{\mathbf{f}}_k = \widetilde{\mathbf{B}}_k^{-1}\mathbf{h}_k \tag{2.33}$$

onde $\widetilde{\mathbf{B}}_k^{-1}$ é a inversa aproximada (ou regularizada) da distorção óptica $\mathbf{B}_k$ [60, 66, 77, 78].

2.3.2.2 *Iterative Back Projection* - IBP

O IBP é um método de super-resolução iterativo que minimiza o erro entre os dados $\mathbf{g}_j$ e a saída do modelo teórico $\mathbf{C}_{j,k}$ [79–81]:

$$\mathbf{f}_k^{n+1} = \mathbf{f}_k^n + \sum_{j=1}^{L} \mathbf{H}_k^{BP} \left(\mathbf{g}_j - \mathbf{C}_{j,k} \mathbf{f}_k^n \right) \tag{2.34}$$

onde n é a iteração corrente e $\mathbf{H}_k^{BP}$ é o operador de *Back Projection*. Os métodos de IBP são similares aos métodos iterativos para solução de mínimos quadrados, como gradiente descendente, Jacobi e Gauss-Seidel [62, 65, 82]. Apesar de convergir rapidamente, o operador de *back projecton* pode divergir ou ser dependente da estimativa inicial. Além disso, o modelo não permite inserir com facilidade informações *a priori* sobre a solução.

2.3.2.3 *Projection Onto Convexs Sets* - POCS

O POCS utiliza todos os modelos e informações *a priori* disponíveis para compor uma série de conjuntos convexos. Idealmente, a imagem de alta resolução $\mathbf{f}_k$ se situa na intersecção de todos os conjuntos convexos. A busca por $\mathbf{f}_k$ é feita por um processo iterativo, cujo resultado é a projeção sobre os conjuntos convexos, conforme:

$$\mathbf{f}_k^{n+1} = \mathcal{P}_q \cdots \mathcal{P}_0 \mathbf{f}_k^n \tag{2.35}$$

onde $\mathcal{P}_m$ é o operador de projeção para o m-ésimo conjunto convexo, assumindo que são usados $q+1$ conjuntos convexos. Este método possui algumas dificuldades, como a determinação dos operadores de projeção, não-unicidade da solução ou múltiplas soluções [62, 65, 72, 83–88].

2.3.2.4 Métodos Determinísticos Regularizados

O modelo a seguir define como solução a imagem que minimiza a discrepância entre os dados e a saída do modelo teórico:

$$\widehat{\mathbf{f}}_k = \underset{\mathbf{f}_k}{\operatorname{argmin}} \, J \left(\mathbf{g}_k - \mathbf{D}_k \mathbf{f}_k \right) \tag{2.36}$$

sendo que $J(\cdot)$ representa cálculo da distância ou discrepância entre dois vetores. As medidas de distância mais utilizadas são: norma ℓ_n, norma Huber, mínimos quadrados, mínimos quadrados ponderados, etc [72]. Neste caso, procura-se uma imagem de alta-resolução $\mathbf{f}_k$ que minimize a distância entre a imagem

em baixa-resolução $\mathbf{g}_k$ e matriz $\mathbf{D}_k$, que representa as distorções inerentes do processo de aquisição da imagem, aplicado a $\mathbf{f}_k$.

Observe que a matriz $\mathbf{D}_k$, de tamanho $N \times M$ e $N < M$, faz com que o sistema seja subdeterminado, pois tem mais incógnitas que equações. Portanto, a Equação 2.36 é um problema mal posto com infinitas soluções.

Uma estratégia utilizada nos algoritmos de super-resolução para o aumento do número de equações do sistema fazendo o uso dos modelos de aquisição e o modelo combinado (de aquisição e movimento de imagens) para restringir o número de soluções possíveis para $\mathbf{f}_k$. Assumindo um conjunto de L imagens capturadas, tem-se:

$$\widehat{\mathbf{f}}_k = \underset{\mathbf{f}_k}{\operatorname{argmin}} \sum_{j=1}^{L} J\left(\mathbf{g}_j - \mathbf{C}_{j,k}\mathbf{f}_k\right). \tag{2.37}$$

Pode-se re-escrever a Equação 2.37 utilizando a seguinte relação $\mathbf{C}_{j,k} = \mathbf{D}_j\mathbf{M}_{j,k}$, de acordo com a Equação 2.25. Note que, quando os subíndices da matriz de transformação do movimento são os mesmos, resulta em uma matriz identidade, ou seja, $\mathbf{C}_{k,k} = \mathbf{D}_k\mathbf{M}_{k,k} = \mathbf{D}_k$. Utilizando a notação da Equação 2.26, temos que:

$$\widehat{\mathbf{f}}_k = \underset{\mathbf{f}_k}{\operatorname{argmin}}\, J\left(\boldsymbol{g} - \mathbf{C}_k\mathbf{f}_k\right) \tag{2.38}$$

Entretanto, as Equações 2.37 e 2.38 ainda podem trazer múltiplos resultados se $LN < M$ ou se o posto da matriz $\mathbf{C}_{j,k}$ for menor que M. Além disso, o sistema é geralmente mal condicionado, o que o torna sensível aos erros de medição.

Para garantir a unicidade e a estabilização da solução pode-se incluir uma penalidade de regularização:

$$\widehat{\mathbf{f}}_k = \underset{\mathbf{f}_k}{\operatorname{argmin}}\, J\left(\boldsymbol{g} - \mathbf{C}_k\mathbf{f}_k\right) + \lambda Q\left(\mathbf{f}_k\right) \tag{2.39}$$

onde $Q\left(\mathbf{f}_k\right)$ é a penalidade de regularização usada para estabilizar a solução do problema (pode-se utilizar o mesmo $Q\left(\mathbf{f}_k\right)$ do modelo de imagem - seção 2.3.1.3) e o λ é o coeficiente que controla a influência da penalidade de regularização. Esta forma de regularização é conhecida como Regulação Generalizada de Tikhonov [62, 72].

2.3.2.5 Métodos Estatísticos Bayesianos

Os métodos Bayesianos são largamente utilizados na super-resolução [89, 90], cujo modelamento é maximizar a probabilidade *a posteriori*:

$$\widehat{\mathbf{f}}_k = \underset{\mathbf{f}_k}{\operatorname{argmax}}\, \rho\left(\mathbf{f}_k|\boldsymbol{g}\right) \tag{2.40}$$

sendo que $\mathbf{f}_k$ é modelado como um vetor aleatório, $\boldsymbol{g}$ como dados de observação de um vetor aleatório e $\widehat{\mathbf{f}}_k$ é a estimativa *maximum a posteriori* (MAP). Em uma abordagem Bayesiana típica, pode se re-escrever a Equação 2.40:

$$\begin{aligned}
\widehat{\mathbf{f}}_k &= \underset{\mathbf{f}_k}{\operatorname{argmax}}\, \rho\left(\mathbf{f}_k|\boldsymbol{g}\right) \\
&= \underset{\mathbf{f}_k}{\operatorname{argmax}}\, \frac{\rho\left(\boldsymbol{g}|\mathbf{f}_k\right)\rho\left(\mathbf{f}_k\right)}{\rho\left(\boldsymbol{g}\right)} \\
&= \underset{\mathbf{f}_k}{\operatorname{argmax}}\, \rho\left(\boldsymbol{g}|\mathbf{f}_k\right)\rho\left(\mathbf{f}_k\right) \\
&= \underset{\mathbf{f}_k}{\operatorname{argmax}}\, \ln\left(\rho\left(\boldsymbol{g}|\mathbf{f}_k\right)\right) + \ln\left(\rho\left(\mathbf{f}_k\right)\right) \\
&= \underset{\mathbf{f}_k}{\operatorname{argmin}}\, -\ln\left(\rho\left(\boldsymbol{g}|\mathbf{f}_k\right)\right) - \ln\left(\rho\left(\mathbf{f}_k\right)\right)
\end{aligned} \tag{2.41}$$

onde nesta forma, as quantidades $\rho\left(\boldsymbol{g}|\mathbf{f}_k\right)$ e $\rho\left(\mathbf{f}_k\right)$ podem ser estimadas por treino, diferentemente da probabilidade *a posteriori* $\rho\left(\mathbf{f}_k|\boldsymbol{g}\right)$. A forma da Equação 2.41 permite chegar a mesma solução do método regularizado determinístico mostrada na Equação 2.39. Basta fazer com que

$$\rho\left(\boldsymbol{g}|\mathbf{f}_k\right) = \frac{1}{j}e^{-\frac{J\left(\boldsymbol{g}-\mathbf{C}_k\mathbf{f}_k\right)}{\theta}} \tag{2.42}$$

e utilizar o modelo de imagem descrito na na Seção 2.3.1.3

$$\rho\left(\mathbf{f}_k\right) = \frac{1}{q}e^{-\frac{Q\left(\mathbf{f}_k\right)}{\beta}}. \tag{2.43}$$

Substituindo as funções densidade de probabilidade das Equações 2.42 e 2.43 na Equação 2.41, obtém-se:

$$\begin{aligned}\widehat{\mathbf{f}}_k &= \underset{\mathbf{f}_k}{\operatorname{argmin}} \frac{J\left(\boldsymbol{g}-\mathbf{C}_k\mathbf{f}_k\right)}{\theta} + \ln(j) + \frac{Q\left(\mathbf{f}_k\right)}{\beta} + \ln(q) \\ &= \underset{\mathbf{f}_k}{\operatorname{argmin}} J\left(\boldsymbol{g}-\mathbf{C}_k\mathbf{f}_k\right) + \frac{\theta}{\beta}Q\left(\mathbf{f}_k\right) \\ &= \underset{\mathbf{f}_k}{\operatorname{argmin}} J\left(\boldsymbol{g}-\mathbf{C}_k\mathbf{f}_k\right) + \lambda Q\left(\mathbf{f}_k\right) \end{aligned} \tag{2.44}$$

onde λ é o coeficiente de regularização definido como $\lambda = \theta/\beta$. Apesar de os métodos determinístico regularizado e estatístico bayesiano resultarem em formulações matematicamente idênticas, conforme mostram as Equações 2.39 e 2.44, eles possuem pontos de vista distintos. Neste caso, $J\left(\boldsymbol{g}-\mathbf{C}_k\mathbf{f}_k\right)$ é chamado de termo de dados e $Q\left(\mathbf{f}_k\right)$ de termo de informação *a priori*.

2.3.2.6 Métodos Baseado em Exemplos

Diferente dos casos anteriores, este modelo se faz o uso de imagens de alta-resolução que são utilizadas como exemplo, relacionadas ou não às imagens a que se deve aplicar a super-resolução. Cada imagem utilizada como exemplo é representada matematicamente por $\mathbf{f}_k$, que passa pelos processos de extração da informação de alta-freqüência $\mathbf{H}_k$ e pela transformação de movimento $\mathbf{M}_{j,k}$. O resultado destas operações é, em seguida, adicionado à imagem de baixa-resolução $\mathbf{g}_j$ interpolada pela matriz $\mathbf{E}_j$. Além disso, um erro $\varepsilon_{j,k}$, associado à transformação de movimento $\mathbf{M}_{j,k}$ e ao processo de extração de alta-freqüência $\mathbf{H}_k$, é representado no modelo. Matematicamente o modelo da super-resolução baseada em exemplos pode ser descrita como:

$$\widehat{\mathbf{f}}_j = \mathbf{E}_j\mathbf{g}_j + \mathbf{M}_{j,k}\mathbf{H}_k\mathbf{f}_k + \varepsilon_{j,k}. \tag{2.45}$$

Existem várias estratégias para compor o conjunto de imagens $\mathbf{f}_k$, que pode ser feita com imagens genéricas, como em [1, 2, 91–94]. Nesta tese, o realce em seqüências de resolução variável é abordado. Diferentemente dos trabalhos mencionados, o conjunto $\mathbf{f}_k$ é obtido dinamicamente ao fazer a busca (estimação de movimento) nos quadros de alta-resolução permitindo obter um dicionário reduzido com grande correlação com a imagem em baixa-resolução.

No trabalho de Freeman *et al.* [91], o MRF é utilizado para relacionar as imagens de baixa-resolução com imagens de alta-resolução. A solução deste modelo foi obtida utilizando o algoritmo de *belief propagation*. Em um trabalho posterior [92] comparou-se o método apresentado em [91] com um algoritmo

que utiliza as fronteiras dos blocos pré-processados para realizar o casamento dos dados do dicionário no processo de super-resolução. Segundo os autores, apesar de ser mais simples, este algoritmo resulta em imagens super-resolvidas com qualidade similar aos baseados em MRF.

Uma outra abordagem é observar o problema de super-resolução com a perspectiva de *compressive sensing* [1, 2, 93, 94]. Neste caso, as imagens de baixa-resolução são vistas como versões subamostradas de uma imagem de alta-resolução, cujas sub-imagens são assumidas tendo uma representação esparsa com respeito a um dicionário composto por muitos elementos.

3 SUPER-RESOLUÇÃO BASEADA EM EXEMPLOS

3.1 INTRODUÇÃO

Na literatura, define-se super-resolução como o processo de aumento de resolução de uma imagem em que se utiliza informações de outras imagens [60, 61, 83]. Por meio do uso de múltiplas imagens correlacionadas, tais métodos podem ultrapassar as limitações inerentes à interpolação quando esta está restrita ao uso de uma única imagem. Em geral, os métodos de super-resolução exploram deslocamentos sub-pixel entre imagens de baixa-resolução para formar uma imagem de alta-resolução. No entanto, alguns métodos de super-resolução baseiam-se em imagens disponíveis de alta-resolução para assim estimar os detalhes que estão ausentes na imagem de baixa-resolução [92, 95]. Por exemplo, Freeman *et al.* [92] usam um conjunto de treinamento composto por imagens de alta-resolução para restaurar as altas freqüências ausentes em imagens sujeitas a *zoom*. De maneira semelhante, a super-resolução é empregada por Brandi *et al.* [95] com vídeo de resolução mista para recuperar quadros de baixa-resolução por meio do uso de informação de alta freqüência presente em quadros vizinhos de alta-resolução. O método de super-resolução apresentado nessa tese assemelha-se a essas duas últimas propostas, pois é também baseado no uso de imagens disponíveis em alta-resolução para realçar imagens de baixa-resolução.

As imagens utilizadas para aumentar a resolução podem ser diferentes fotos da mesma cena, diferentes quadros do mesmo vídeo, ou podem simplesmente ser compostas por um banco de dados formado por imagens que aparentemente não guardam relação com as imagens originais. A interpolação difere da super-resolução por utilizar apenas as informações dos pixels vizinhos para estimar os pixels "faltantes". Ou seja, na interpolação, a estrutura da informação local dita como a informação "faltante" é preenchida, portanto os métodos de interpolação raramente introduzem informações de alta freqüência.

Contudo, a super-resolução se utiliza da informação de diferentes imagens do mesmo objeto, ou conteúdos similares para inferir que informação de alta freqüência está faltando. Como a super-resolução é mais "arrojada" que a interpolação, a mesma é capaz de recuperar ou introduzir informações de alta freqüência, enquanto corre o risco de introduzir artefatos espúrios durante o processo.

Na super-resolução baseada em exemplos, um banco de dados com imagens de referência $\boldsymbol{f} = \left[\mathbf{f}_1^T \cdots \mathbf{f}_L^T\right]^T$ é associado às suas versões de baixa-resolução $\boldsymbol{f_L} = \left[\mathbf{f_L}_1^T \cdots \mathbf{f_L}_L^T\right]^T$. Para uma dada imagem com resolução menor $\mathbf{g}_k$, uma interpolação $\mathbf{g_I}_k = \mathbf{E}_k\mathbf{g}_k$ antes de se fazer uma busca em $\boldsymbol{f_L}$. Quando um

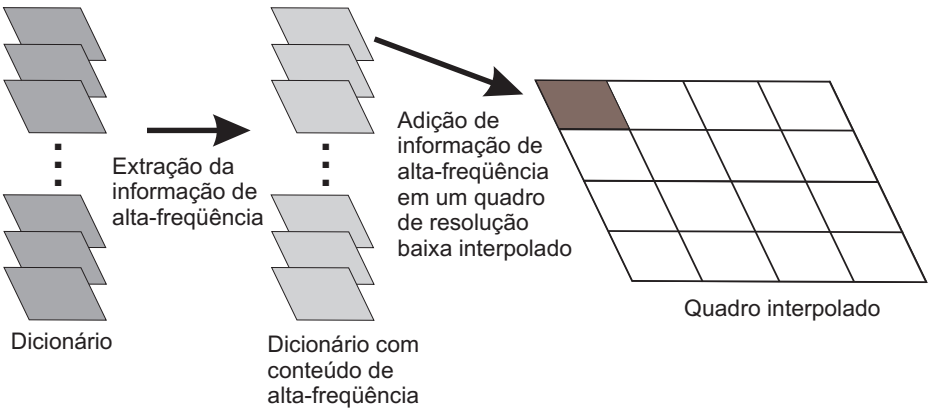


Figura 3.1: Diagrama geral da super-resolução baseada em exemplos.

bom casamento entre $\mathbf{g}_{\mathbf{I}_k}$ $\mathbf{f}_{\mathbf{L}\nu}$ é encontrado, sua informação de alta-freqüência ($\mathbf{f}_{\mathbf{H}\nu} = \mathbf{H}_\nu \mathbf{f}_\nu$) é aplicada em $\mathbf{g}_{\mathbf{I}_k}$. Entretanto, a utilização de imagens inteiras no banco de dados $\boldsymbol{f_L}$ dificultam o casamento com $\mathbf{g}_{\mathbf{I}_k}$ devido à complexidade do conteúdo das imagens. Por isso, os processos de super-resolução geralmente utilizam blocos ou porções da imagem de tamanhos retangulares para os processos de modelamento, treinamento e realce. Nesta tese, serão utilizadas as notações $\dot{\boldsymbol{f}} = \left[\dot{\mathbf{f}}_1^T \cdots \dot{\mathbf{f}}_L^T\right]^T$ e $\dot{\mathbf{g}}_k$ para representar os blocos do banco de dados de referência e os blocos em baixa-resolução, respectivamente.

3.2 DESCRIÇÃO DO MÉTODO DE SUPER-RESOLUÇÃO BASEADA EM EXEMPLOS

Nesta seção será detalhado o algoritmo de super-resolução baseada em exemplos [21, 95–97] utilizado nesta tese. A forma geral da super-resolução baseada em exemplos é ilustrada na Figura 3.1. Nela uma imagem é dividida em retângulos ou blocos de $N_1 \times N_2$ pixels. Assuma que cada bloco k seja representado por um vetor $\dot{\mathbf{g}}_k$ de tamanho $N \times 1$, onde $N = N_1 N_2$, e que se necessite de um aumento de resolução por um fator $s > 1$. Desta forma, cada bloco realçado $\widehat{\dot{\mathbf{g}}_{\mathbf{R}k}}$ deverá ter $s^2 N \times 1$ pixels e uma informação de alta-freqüência $\dot{\mathbf{f}}_{\mathbf{H}_\nu}$ deverá ser adicionada à versão interpolada de $\dot{\mathbf{g}}_k$, $\dot{\mathbf{g}}_{\mathbf{I}_k}$, ou seja: $\widehat{\dot{\mathbf{g}}_{\mathbf{R}k}} = \dot{\mathbf{f}}_{\mathbf{H}_\nu} + \mathbf{E}_k \dot{\mathbf{g}}_k$.

Um banco de dados $\dot{\boldsymbol{f}}$ é populado por blocos "exemplos" $\dot{\mathbf{f}}_i$ de $s^2 N \times 1$ pixels, gerados a partir de várias imagens de referência. $\dot{\boldsymbol{f}}$ pode ser muito grande, contendo centenas ou até milhares de blocos-exemplos.

Cada bloco-exemplo $\dot{\mathbf{f}}_i$ é filtrado por um filtro passa-baixas $\mathbf{F}_0$, resultando em $\dot{\mathbf{f}}_{\mathbf{L}_i} = \mathbf{F}_0(\dot{\mathbf{f}}_i)$ e sua respectiva versão em passa-altas pode ser obtida por $\dot{\mathbf{f}}_{\mathbf{H}_i} = \dot{\mathbf{f}}_i - \dot{\mathbf{f}}_{\mathbf{L}_i}$. O filtro $\mathbf{F}_0$ é composto pelos processos encadeados de redução e interpolação de imagens, conforme a Seção 2.2.

O processo de super-resolução funciona da seguinte forma. O bloco $\dot{\mathbf{g}}_k$ a ser super-resolvido é interpolado para gerar $\dot{\mathbf{g}}_{\mathbf{I}_k}$ que é comparado com cada $\dot{\mathbf{f}}_{\mathbf{L}_i}$ em alguma métrica de distância D e em seguida faz-se uma busca para minimizar a seguinte relação: $\nu = \underset{i}{\operatorname{argmin}}\ D(\dot{\mathbf{g}}_{\mathbf{I}_k}, \dot{\mathbf{f}}_{\mathbf{L}_i})$, *i. e.* $\dot{\mathbf{f}}_{\mathbf{L}_\nu}$ é escolhido como melhor casamento (*matching*).

A informação de alta-freqüência vinculada a $\dot{\mathbf{f}}_{\mathbf{L}_\nu}$ é $\dot{\mathbf{f}}_{\mathbf{H}_\nu}$, à qual relacionamos a informação de alta-freqüência desejada do bloco $\dot{\mathbf{g}}_k$, ou seja, consideramos $\overline{\dot{\mathbf{g}}_{\mathbf{H}k}} \cong \dot{\mathbf{f}}_{\mathbf{H}_\nu}$ e o bloco realçado é obtido por: $\widehat{\dot{\mathbf{g}}_{\mathbf{R}k}} = \dot{\mathbf{f}}_{\mathbf{H}_\nu} + \dot{\mathbf{g}}_{\mathbf{I}_k}$. Entretanto, este método pode adicionar informações de alta-freqüência espúrias ao invés de realçar o bloco. Portanto, popular o dicionário com conteúdos correlacionados é crucial para um bom desempenho do algoritmo. Por tratarmos de realce de vídeo, utilizaremos a estimação de movimento entre os quadros-chave e não-chave para compor um dicionário com grande correlação espacial e dinamicamente populado. Além disso, combinar o conteúdo de dicionários compostos por diversas fontes tende a uma melhoria nos resultados. Nesta tese, será proposta uma técnica para reduzir significativamente os erros no processo de realce mediante a combinação de múltiplos resultados dos bancos de exemplos.

3.2.1 Combinação de informações a partir de múltiplos dicionários

Sejam K dicionários populados utilizando-se fontes diferentes ou imagens de referência com diferentes características. Seja o n-ésimo dicionário $\dot{\boldsymbol{f}}_{(n)}$ contendo blocos $\dot{\mathbf{f}}_{i(n)}$ com suas respectivas versões em passa-baixas $\dot{\mathbf{f}}_{\mathbf{L}_{\nu(n)}}$ e passa-altas $\dot{\mathbf{f}}_{\mathbf{H}_{\nu(n)}}$. Seja ainda $\nu(n)$ o índice do bloco com o melhor casamento do n-ésimo dicionário, onde cada dicionário é populado com blocos de uma janela de busca de um quadro de referência. Busca-se uma combinação linear das informações dos dicionários dada por:

$$\widehat{\dot{\mathbf{g}}_{\mathbf{H}k}} = \sum_{n=1}^{K} \alpha_n \dot{\mathbf{f}}_{\mathbf{H}_{\nu(n)}}. \tag{3.1}$$

Para calcular α_n, seja $\widehat{\dot{\mathbf{g}}_{\mathbf{H}k}}$ a camada de realce (bloco com a informação de alta-freqüência) de um bloco estimado a partir da fusão de múltiplas informações e seja $\dot{\mathbf{f}}_{\mathbf{H}_{\nu(n)}}$ uma predição do bloco de realce da n-ésima referência, cujo dicionário pode ser populado a partir de diferentes referências, como quadros posteriores, anteriores ou imagens de uma mesma cena. Seja também $\overline{\dot{\mathbf{g}}_{\mathbf{H}k}}$ o realce ideal de um bloco e $\boldsymbol{\zeta}_n$ o erro espacial do melhor casamento do n-ésimo dicionário. O bloco de realce predito pode ser modelado por

$$\dot{\mathbf{f}}_{\mathbf{H}_{\nu(n)}} = \overline{\dot{\mathbf{g}}_{\mathbf{H}k}} + \boldsymbol{\zeta}_n, \quad \boldsymbol{\zeta}_n \sim N(\mathbf{0}, \boldsymbol{Z}_n), \tag{3.2}$$

assumindo que os ruídos ($\boldsymbol{\zeta}_n$) são independentes (*i.i.d.*).

Seja $\dot{\boldsymbol{f}}_{\boldsymbol{H}\nu} = [\dot{\mathbf{f}}^T_{\mathbf{H}_{\nu(1)}}, ..., \dot{\mathbf{f}}^T_{\mathbf{H}_{\nu(K)}}]^T$ o conjunto de bons casamentos obtidos em K dicionários que estimam o bloco de realce ideal $\overline{\dot{\mathbf{g}}_{\mathbf{H}k}}$. Assumimos que a função densidade de probabilidade (PDF, do inglês, *probability density function*) de $\overline{\dot{\mathbf{g}}_{\mathbf{H}k}}$ possa ser modelada por uma distribuição Gaussiana com média $\boldsymbol{\mu}_0$ e matriz de covariância $\boldsymbol{Z}_0$. A PDF de um bloco predito de realce é condicionada ao bloco ideal da seguinte forma: $\rho(\dot{\boldsymbol{f}}_{\boldsymbol{H}\nu}|\overline{\dot{\mathbf{g}}_{\mathbf{H}k}})$, que resulta em uma Gaussiana com média $\boldsymbol{\mu}_0$ e uma matriz de covariância $diag[\boldsymbol{Z}_1, \boldsymbol{Z}_2, ..., \boldsymbol{Z}_K]$.

Já $\rho(\dot{\boldsymbol{f}}_{\boldsymbol{H}\nu})$ é dado pela seguinte relação:

$$\rho(\dot{\boldsymbol{f}}_{\boldsymbol{H}\nu}) = \int_{-\infty}^{\infty} \rho(\dot{\boldsymbol{f}}_{\boldsymbol{H}\nu}|\overline{\dot{\mathbf{g}}_{\mathbf{H}k}})\rho(\overline{\dot{\mathbf{g}}_{\mathbf{H}k}})d\overline{\dot{\mathbf{g}}_{\mathbf{H}k}}, \tag{3.3}$$

ao resolver a Equação 3.3 obtém-se uma função Gaussiana com média $\boldsymbol{\mu}_0$ e matriz de covariância $\sum_{n=1}^{K} \boldsymbol{Z}_n + \boldsymbol{Z}_0$. A PDF *a posteriori* de $\overline{\dot{\mathbf{g}}_{\mathbf{H}k}}$, dado a informação predita $\dot{\boldsymbol{f}}_{\boldsymbol{H}\nu}$, *i. e.* $\rho(\overline{\dot{\mathbf{g}}_{\mathbf{H}k}}|\dot{\boldsymbol{f}}_{\boldsymbol{H}\nu})$ é obtida pelo Teorema de Bayes:

$$\rho(\overline{\dot{\mathbf{g}}_{\mathbf{H}k}}|\dot{\boldsymbol{f}}_{\boldsymbol{H}\nu}) = \rho(\dot{\boldsymbol{f}}_{\boldsymbol{H}\nu}|\overline{\dot{\mathbf{g}}_{\mathbf{H}k}})\rho(\overline{\dot{\mathbf{g}}_{\mathbf{H}k}})\rho(\dot{\boldsymbol{f}}_{\boldsymbol{H}\nu})^{-1} \tag{3.4}$$

também resulta em uma Gaussiana, mas com média $\left(\sum_{n=1}^{K} \boldsymbol{Z}_n^{-1} + \boldsymbol{Z}_0^{-1}\right)^{-1} \left(\sum_{n=1}^{K} \dot{\mathbf{f}}_{\mathbf{H}_{\nu(n)}} \boldsymbol{Z}_n^{-1} + \boldsymbol{\mu}_0 \boldsymbol{Z}_0^{-1}\right)$ e matriz de covariância $\left(\sum_{n=1}^{K} \boldsymbol{Z}_n^{-1} + \boldsymbol{Z}_0^{-1}\right)^{-1}$. Uma maneira de se fundir estas predições seria utilizar um critério de máximo *a posteriori* (MAP)

$$\widehat{\dot{\mathbf{g}}_{\mathbf{H}k}} = \underset{\overline{\dot{\mathbf{g}}_{\mathbf{H}k}}}{\operatorname{argmax}} \left(\ln\left(\rho(\overline{\dot{\mathbf{g}}_{\mathbf{H}k}}|\dot{\boldsymbol{f}}_{\boldsymbol{H}\nu})\right)\right). \tag{3.5}$$

Na Equação 3.5 a estimação da informação fundida pelo MAP é dada pela maximização da probabilidade de $\overline{\dot{\mathbf{g}}_{\mathbf{H}k}}$ dadas as informações passa-altas dos dicionários $\dot{\boldsymbol{f}}_{\boldsymbol{H}\nu}$ que, para uma distribuição Gaussiana, resulta na média *a posteriori*:

$$\widehat{\dot{\mathbf{g}}_{\mathbf{H}k}} = \left(\sum_{n=1}^{K} \dot{\mathbf{f}}_{\mathbf{H}_{\nu(n)}} \boldsymbol{Z}_n^{-1} + \boldsymbol{\mu}_0 (\boldsymbol{Z}_0)^{-1}\right) \left(\sum_{n=1}^{K} \boldsymbol{Z}_n^{-1} + (\boldsymbol{Z}_0)^{-1}\right)^{-1}. \tag{3.6}$$

Para obtermos uma estimativa utilizando máxima verossimilhança (ML, do inglês, *maximum likelihood*) a partir do MAP, deve-se assumir uma *a priori* não informativa que pode ser realizada ao assumir $\sigma_0^2 \to \infty$, onde $\boldsymbol{Z}_0 = \mathbf{I}\sigma_0^2$, onde $\mathbf{I}$ é a matriz identidade, chegando à forma:

$$\widehat{\dot{\mathbf{g}}_{\mathbf{H}k}} = \left(\sum_{n=1}^{K} \dot{\mathbf{f}}_{\mathbf{H}_{\nu(n)}} \boldsymbol{Z}_n^{-1}\right) \left(\sum_{n=1}^{K} \boldsymbol{Z}_n^{-1}\right)^{-1} \tag{3.7}$$

Observe que na Equação 3.7 as matrizes de covariância $\boldsymbol{Z}_n$ estão relacionadas com a confiança na predição da informação de alta-freqüência. Entretanto esta informação não é facilmente medida ou estimada. Neste tese propõe-se utilizar a distorção $D_n = D_{SSD}(\dot{\mathbf{g}}_{\mathbf{I}_k}, \dot{\mathbf{f}}_{\mathbf{L}_n})$ baseada na SSD de forma que possamos medir a distância entre blocos dos quadros-não-chave ($\dot{\mathbf{g}}_{\mathbf{I}_k}$) e dos quadros-chave filtrado ($\dot{\mathbf{f}}_{\mathbf{L}_n}$). Ao utilizar D_n em detrimento de $\boldsymbol{Z}_n$ pode-se escrever a Equação 3.7 como:

$$\widehat{\dot{\mathbf{g}}_{\mathbf{H}k}} = \left(\sum_{n=1}^{K} \frac{\dot{\mathbf{f}}_{\mathbf{H}_{\nu(n)}}}{D_n} \right) \left(\sum_{n=1}^{K} \frac{1}{D_n} \right)^{-1}. \tag{3.8}$$

Finalmente, α_n é calculado pela seguinte relação:

$$\alpha_n = \left(\frac{1}{D_n} \right) \left(\sum_{n=1}^{K} \frac{1}{D_n} \right)^{-1}. \tag{3.9}$$

Na Equação 3.9 os pesos de cada bloco de predição das informações de alta-freqüência $\dot{\mathbf{f}}_{\mathbf{H}_{\nu(n)}}$ são calculados a partir de um conjunto de K dicionários. O termo $1/D_n$ implica que o peso relativo à $\dot{\mathbf{f}}_{\mathbf{H}_{\nu(n)}}$ é inversamente proporcional à distorção D_n, sendo em seguida normalizada pelo termo $\left(\sum_{n=1}^{K} 1/D_n \right)^{-1}$. Durante o processo de fusão dos melhores candidatos a blocos com conteúdo de alta-freqüência, se $D_{SSD}(\dot{\mathbf{g}}_{\mathbf{I}_k}, \dot{\mathbf{f}}_{\mathbf{L}_n}) \gg D_{SSD}(\dot{\mathbf{g}}_{\mathbf{I}_k}, \dot{\mathbf{f}}_{\mathbf{L}_m})$, então por conseqüência $\alpha_n \ll \alpha_m$. Se isto ocorre para todos os blocos de uma imagem, então pode-se afirmar que o n-ésimo dicionário foi completamente dominado pelo m-ésimo, tornando-o irrelevante.

Em suma, para se criar os dicionários, selecione os blocos $\dot{\mathbf{f}}_{i(n)}$, onde $1 \leq i \leq L$, $1 \leq n \leq K$ e para cada bloco separe as informações de baixa-freqüência $\dot{\mathbf{f}}_{\mathbf{L}_{i(n)}} = \mathbf{F}_2(\mathbf{F}_1(\dot{\mathbf{f}}_{i(n)}))$ e alta-freqüência $\dot{\mathbf{f}}_{\mathbf{H}_{i(n)}} = \dot{\mathbf{f}}_{i(n)} - \dot{\mathbf{f}}_{\mathbf{L}_{i(n)}}$, onde $\mathbf{F}_1$ e $\mathbf{F}_2$ são filtros de redução e interpolação de imagens, respectivamente.

Para aumentar a resolução de um bloco, propõe-se:

- O bloco que sofrerá aumento de resolução $\dot{\mathbf{g}}_k$ deve ser interpolado por um fator s para se obter $\dot{\mathbf{g}}_{\mathbf{I}_k}$.
- Para cada dicionário n procure por: $\nu(n) = \underset{n}{\operatorname{argmin}}\, D(\dot{\mathbf{g}}_{\mathbf{I}_k}, \dot{\mathbf{f}}_{\mathbf{L}_{i(n)}})$
- Resolva $\{\alpha_n\}$ utilizando a Equação 3.9, onde $D_n = D(\dot{\mathbf{g}}_{\mathbf{I}_k}, \dot{\mathbf{f}}_{\mathbf{L}_{i(n)}})$.
- Obtenha $\widehat{\dot{\mathbf{g}}_{\mathbf{H}k}}$ por meio da Equação 3.8.
- O bloco com o aumento de resolução é dado por $\widehat{\dot{\mathbf{g}}_{\mathbf{R}k}} = \dot{\mathbf{g}}_{\mathbf{I}_k} + \widehat{\dot{\mathbf{g}}_{\mathbf{H}k}}$.

3.2.2 Compensação de blocos multi-escala com sobreposição

Nos processos de codificação de vídeo mais recentes, como o H.264, uma análise em termos de taxa-distorção é realizada de forma a escolher os modos de predição e partição [4,5,98]. Para tanto, uma função de custo J dada por $J = D + \lambda R$, que controla a influência entre D e R, é utilizada para definir o tipo de codificação e a partição de um determinado bloco. Entretanto, o realce é processado no decodificador, sendo possível calcular apenas a distorção entre os blocos $\dot{\mathbf{g}}_{\mathbf{I}_k}$ e $\dot{\mathbf{f}}_{\mathbf{L}_{i(n)}}$. No entanto, ao computarmos apenas a distorção, temos que a estimação de movimento utilizando macroblocos de 16×16 pixels é um sub-conjunto (com distorção menor ou igual que os sub-macroblocos correspondentes) da mesma operação com 8×8 pixels. Apesar do processo de casamento gerar menor distorção, os resultados mostrados em [95] mostram que blocos de 16×16 pixels obtêm, no geral, melhores resultados. Diferentemente da estimação de movimento realizada durante o processo de codificação, não estamos interessados apenas no erro de predição, mas também na detecção dos objetos em cena que necessitem de realce. Apesar de que em blocos de tamanhos maiores, como de 16×16, a estrutura dos objetos são mais fáceis de serem identificadas do que em blocos particionados. Entretanto, ao utilizarmos blocos de tamanho menores, de 8×8 pixels, podemos realçar com um detalhamento maior. Contudo, aumentam as chances de ocorrerem descasamentos (*mismatches*) entre os conteúdos do dicionário. Portanto, propôs-se a utilização de vários tamanhos de bloco para o realce na tentativa de somar as vantagens da utilização de cada tamanho de bloco diferente. Nesta tese sugere-se um fator de penalização, que multiplica a distorção calculada no processo de estimação de movimento dos blocos particionados co-localizados. A Figura 3.2 mostra o comportamento da super-resolução após a variação do fator de penalização para a seqüência *Shields*. Cada curva da figura é o resultado de um quadro realçado entre os quadros-chave 1 e 31. Para cada curva variou-se o fator de penalização aplicado à partição de blocos, onde o custo de quatro blocos co-localizados de 8×8 multiplicado pelo fator de penalização é comparado com o bloco de 16×16. O bloco, particionado ou não particionado, com menor custo é escolhido para a super-resolução. Observe que a PSNR de cada curva foi normalizada pelo seu valor máximo, sendo assim possível verificar que o fator de penalização gera bons resultados entre os valores 1,3 e 2,2.

As técnicas tradicionais de estimação e compensação de movimento geram ruídos de blocos, que são gerados pelos "descasamentos" entre as fronteiras dos blocos. Neste trabalho, propõe-se a utilização da compensação de movimentos utilizando blocos sobrepostos (do inglês, *overlapped block motion compensation* − OBMC) em adição aos blocos de tamanhos variados [14,21,99,100]. Em [100], propôs-se um "re-particionamento virtual" dos blocos para que blocos de diferentes tamanhos possam ser aplicados

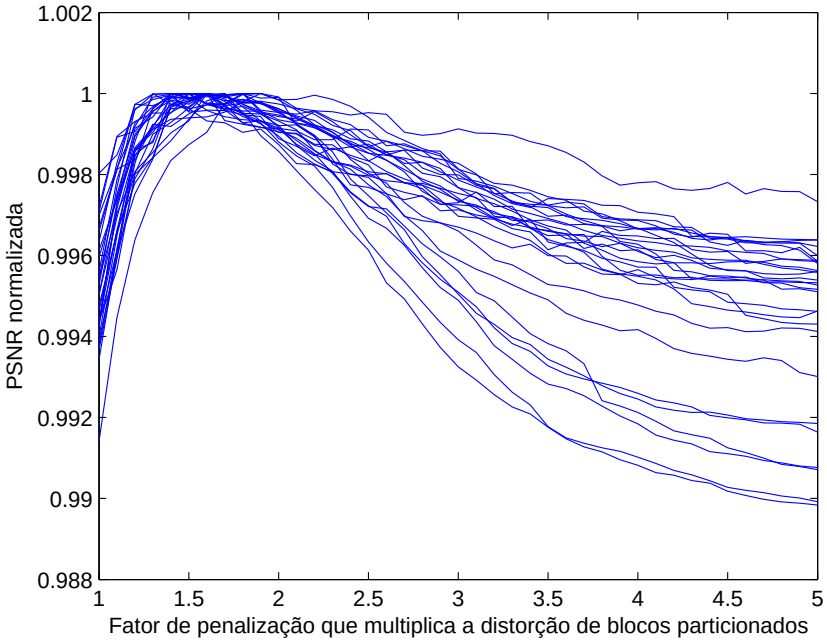


Figura 3.2: Desempenho do realce ao variar o fator de penalização aplicado à partição de blocos, onde o custo de quatro blocos co-localizados de 8×8 multiplicado pelo fator de penalização é comparado com o bloco de 16×16. O bloco, particionado ou não particionado, com menor custo é escolhido para a super-resolução. Nestas curvas, a PSNR foi normalizada pelo valor máximo de cada curva.

na OBMC. Neste caso, os blocos são particionados até que o menor tamanho seja utilizado, o que permite um esquema equivalente ao OBMC utilizando blocos de tamanho fixo, como ilustrado na Figura 3.3.

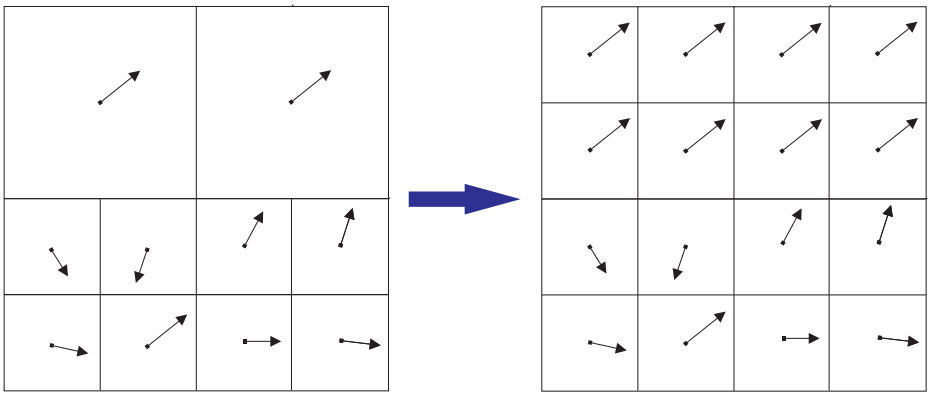


Figura 3.3: Processo onde os blocos de tamanho variáveis com 16×16 e 8×8 pixels são "virtualmente re-particionados" de forma que possam ser aplicados a compensação de movimento com sobreposição (OBMC).

A Figura 3.4 ilustra os blocos sobrepostos na OBMC. Neste caso, propomos utilizar apenas uma sobreposição de dois pixels, pois mantém a informação de alta-freqüência no centro dos blocos intacta. Além disso, também reduz-se as diferenças entre a utilização de uma estimação de movimento que leva em consideração a sobreposição e uma estimação de movimentos em blocos ordinários, permitindo que a OBMC proposta seja compatível com os algoritmos rápidos de estimação de movimento [39–44].

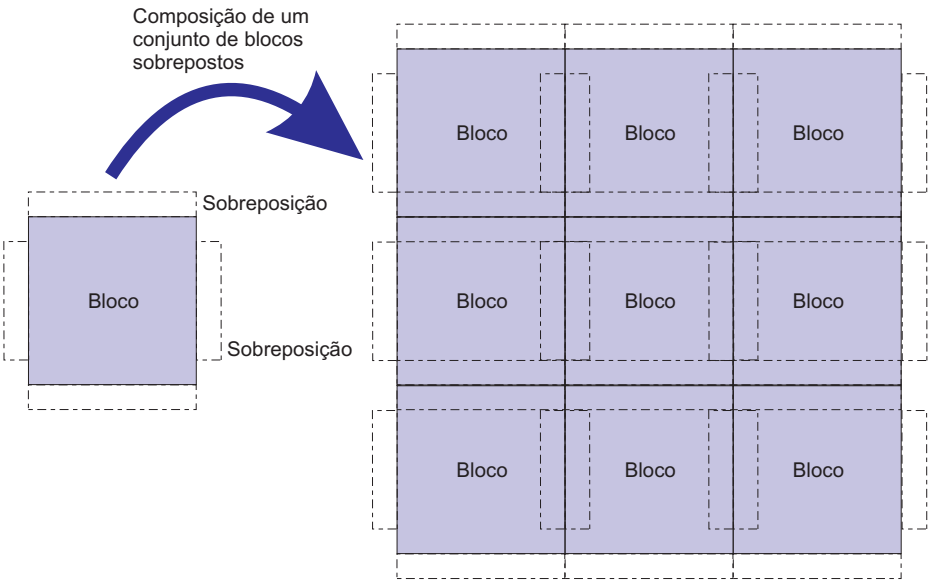


Figura 3.4: Sobreposição dos blocos utilizados na OBMC.

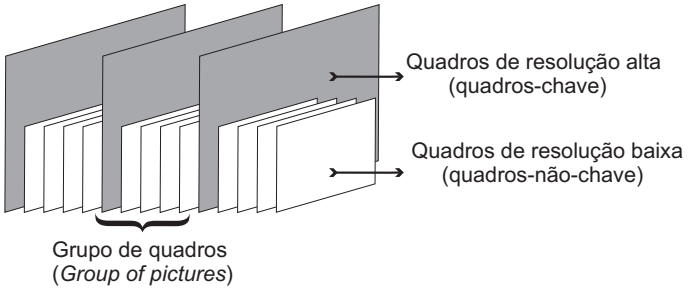


Figura 3.5: Formato do vídeo composto por quadros de resolução mista.

3.3 CODIFICAÇÃO DE VÍDEO COM RESOLUÇÃO MISTA

Na super-resolução baseada em exemplos, uma composição de dicionários com conteúdos correlacionados tende a melhorar os resultados da técnica proposta. Basicamente, bons exemplos geram bons pares em baixa-resolução, o que permite uma boa associação aos conteúdos de alta-resolução, permitindo assim bons resultados. Ao se aplicar a técnica de ponderação dos vários dicionários baseada na métrica de distorção proposta nesta tese, pode-se verificar que alguns dicionários são "dominantes" se comparados com outros, o que permite a utilização de bons dicionários em detrimentos dos menos correlacionados. A super-resolução de vídeo baseada em exemplos utiliza imagens de alta-resolução correlacionadas para realçar a resolução do vídeo. De fato, em algumas aplicações, quadros ou imagens de resolução maior que o vídeo podem estar disponíveis, obtendo assim, resultados na super-resolução mais interessantes que dicionários pré-definidos ou treinados *offline*.

3.3.1 Codificação de vídeo composto por quadros de resolução mista

Na codificação onde os quadros de vídeo possuem resolução mista, podemos associar os quadros de resolução maior aos quadros-chave e os de resolução menor aos quadros-não-chave. Esta codificação permite uma redução da quantidade de bits ou ainda a redução da complexidade do codificador [16, 17].

A Figura 3.5 ilustra um vídeo codificado, onde os quadros-chaves são periodicamente intercalados entre os quadros-não-chave. Portanto, se o período de quadro-chaves (conhecido também por GOP, do inglês *group of pictures*) é G, então para cada quadro-não-chave a ser realçado existe um quadro-chave com diferenças temporais de no máximo $G/2$ quadros. Esta informação é interessante para definir o valor de G a ser utilizado, pois este parâmetro controla indiretamente a similaridade entre quadros, visto que tipicamente, quanto mais próximos os quadros-chave dos não-chave, maior a similaridade entre eles. Dependendo da aplicação, pode-se ter conteúdos com muito movimento, como por exemplo uma partida

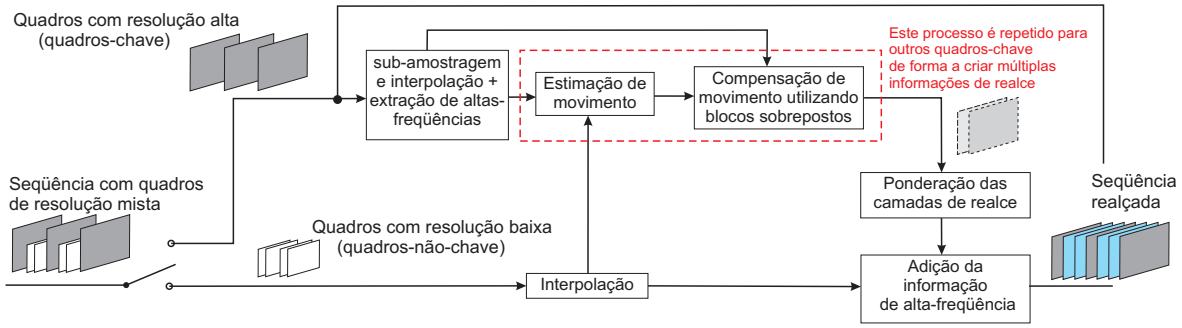


Figura 3.6: Diagrama de blocos para a super-resolução de vídeos compostos por quadros de resolução mista.

de futebol, necessitando assim de um valor de G pequeno, ou poucos movimentos, como em telejornais, permitindo assim um valor de G maior.

A Figura 3.6 ilustra o diagrama de blocos que representa o processo de super-resolução de vídeos compostos por quadros de resolução mista utilizando os quadros de resolução maior (quadros-chaves). Nele, a primeira etapa é distinguir entre os quadros-chave e não-chave. Os quadros-chave são sub-amostrados e interpolados utilizando o filtro Lanczos [59], gerando assim uma versão de baixa-freqüência do quadro-chave. Em seguida fazemos uma estimação de movimento bi-direcional entre as versões interpoladas dos quadros-chave e o quadro não-chave.

Note que o processo de estimação de movimento dinamicamente popula um dicionário com conteúdos originados nos quadros-chave e que, no geral, correspondem à informação nos quadros-não-chave a serem realçadas. A estimação de movimento permite a redução do tamanho do dicionário ao utilizar a informação da posição espacial (i, j) do bloco atual no quadro-não-chave, além de evitar a busca em toda a imagem ou em várias imagens. Ao fazer uma busca completa em uma janela de $w_w \times w_h$ pixels em torno de (i, j) nos quadros-chave, realiza-se $w_w w_h$ comparações, ao invés de wh comparações para cada quadro-chave de $w \times h$ pixels . Os tamanhos das janelas de estimação são, no geral, valores da ordem de $w_w = w/8$ e/ou $w_h = h/8$, o que reduz o número de blocos do dicionário em algumas ordens de grandeza. Neste cenário, outros métodos de estimação de movimento rápidas [39–44] também podem ser utilizadas para a redução da complexidade, ou redução do tempo de processamento. A estimação de movimento é feita utilizando tamanho de blocos variáveis (16 $\times$ 16- e 8 $\times$ 8 pixels). A compensação de blocos com sobreposição utiliza os vetores de movimento do processo de estimação de movimento juntamente com a informação de alta-freqüência extraída dos quadros-chave, permitindo a criação de uma camada contendo as informações

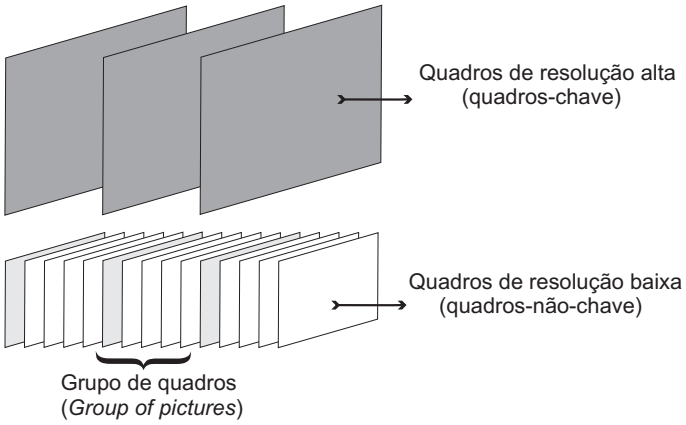


Figura 3.7: Formato de vídeo com fotos redundantes.

de alta-freqüência registradas e compatíveis com o conteúdo de baixa-resolução. Os quadros-não-chave realçados são obtidos ao se adicionar a camada de alta-freqüência ao quadro interpolado.

3.3.2 Vídeo com fotos redundantes

Em outro cenário de aplicação, a câmera captura e comprime os vídeos em baixa-resolução, mas tira fotografias periódicas em alta-resolução, por exemplo, uma imagem em JPEG por segundo, como ilustra a Figura 3.7. Neste caso, as imagens em alta-resolução popularão os dicionários e servirão como quadros-chave, permitindo também o uso da estimação de movimentos para explorar as redundâncias temporais e diminuir os número de comparações nos dicionários. Diferentemente da aplicação anterior, obtemos quadros-chave e não-chave redundantes capturados no mesmo instante de tempo, o que simplifica a extração das informações de alta-freqüência de um quadro-chave. Entretanto, o quadro redundante gera um acréscimo na taxa de bits do vídeo em questão. Além disso, utilizamos dois padrões de codificação diferentes: um codificador de vídeo e um codificador de imagens.

Na Figura 3.8, o processo de realçar o vídeo baseado em fotografias redundantes é ilustrado. Primeiramente, devem-se associar as fotografias aos quadros-não-chave que foram simultaneamente capturados. Assim, como na aplicação anterior, a estimação de movimento é feita utilizando os quadros-chave e os não-chave interpolados. Entretanto, a extração da alta-freqüência é feita a partir da diferença entre a fotografia e o quadro-não-chave (capturada no mesmo instante que a foto) interpolado. Esta informação é compensada em blocos com sobreposição a partir dos vetores de movimento. Por fim, o vídeo realçado é obtido ao se adicionar a camada de alta-freqüência ao quadro-não-chave interpolado.

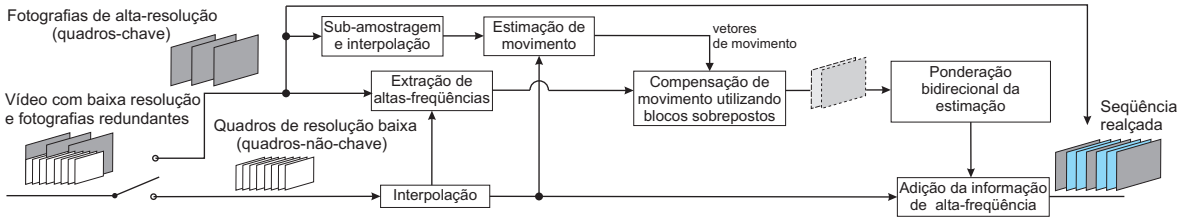


Figura 3.8: Arquitetura do realce baseado em exemplos aplicado à uma seqüência de vídeo com fotografias simultâneas.

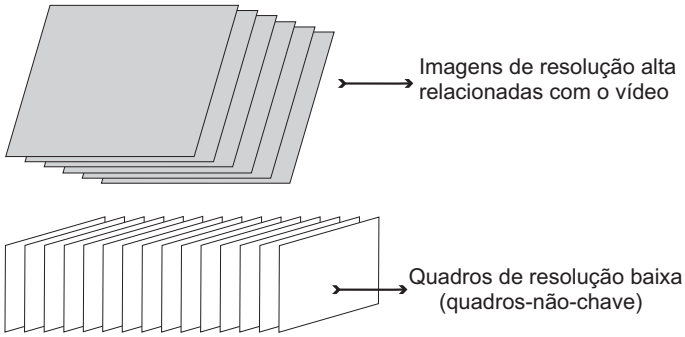


Figura 3.9: Vídeo em baixa-resolução com banco de dados em alta-resolução contendo fotografias correlacionadas. A super-resolução seria aplicada ao vídeo utilizando o banco de dados como exemplos.

3.3.3 Outros cenários de aplicação

Uma outra aplicação possível para o realce seria aumentar a resolução do vídeo comprimido. Este processo pode ser feito *offline* baseado em um banco de fotos com cenas similares, como ilustra a Figura 3.9. Esta aplicação equivale à aplicar a super-resolução baseada em exemplos, proposta em [92], quadro a quadro. Note que as fotografias utilizadas para popular o dicionário não definem um GOP, como em cenários previamente descritos, o que limita a utilização da estimação de movimento. Além disso, as fotografias deve ser bem selecionadas para que o método seja mais eficiente. Como exemplo, podemos utilizar imagens obtidas da mesma posição cartesiana, geodésica e/ou altimétrica. Todavia, outros critérios também podem ser adotados, como a energia da imagem, histograma, histograma das cores, etc. Contudo, os métodos de super-resolução podem adicionar erros de alta-freqüência aos vídeos realçados.

Por fim, outro cenário de aplicação, ilustrado na Figura 3.10, é aplicado em ocultamento de erros (conhecido na literatura como *error concealment*). Neste caso, um vídeo de resolução alta é comprimido e enviado por um canal ruidoso, enquanto uma mesma versão do vídeo é enviada em outro canal não-ruidoso. Assim, o vídeo de baixa-resolução é utilizado apenas quando existem erros na transmissão do vídeo em alta-resolução. Os quadros de alta-resolução que sofreram erros de transmissão são substituídos pelos de baixa-resolução, que em seguida são realçados para aumentar a qualidade (resolução) do vídeo. [22].

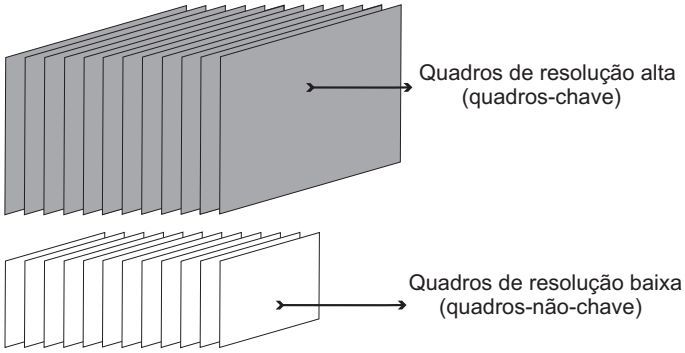


Figura 3.10: Vídeo de alta-resolução com quadros redundantes de baixa-resolução aplicados em ocultamento de erros. Onde um vídeo de resolução alta é comprimido e enviado por um canal ruidoso, enquanto uma mesma versão do vídeo é enviada em outro canal não-ruidoso. Assim, o vídeo de baixa-resolução é utilizado apenas quando existem erros na transmissão do vídeo em alta-resolução. Os quadros de alta-resolução que sofreram erros de transmissão são substituídos pelos de baixa-resolução, que em seguida são super-resolvidos utilizando os quadros sem erros do vídeo em alta-resolução.

3.4 RESULTADOS EXPERIMENTAIS DA SUPER-RESOLUÇÃO EM VÍDEOS COM RESOLUÇÃO MISTA

O desempenho dos métodos de super-resolução baseado em exemplos é determinado, principalmente, pela correlação entre os quadros de baixa-resolução com as imagens exemplos de resolução maior. Por exemplo, realizamos um teste onde 300 quadros da seqüência *Foreman*, originalmente no formato CIF (352×288 pixels) foram reduzidas para QCIF (176×144 pixels) utilizando o método bilinear. Ao interpolar os quadros de baixa-resolução para o tamanho original, utilizando a técnica bilinear, obtém-se a PSNR média de 28,97 dB. Entretanto ao aplicar a super-resolução utilizando a imagem *Lena* (de tamanho 512×512 pixels) como dicionário atinge-se apenas uma PSNR de média 29,01 dB. Todavia, os resultados a seguir utilizam como exemplo informações bastante correlacionadas, obtendo assim resultados mais significativos.

Inicialmente, será apresentado o desempenho isolado de algumas técnicas propostas nesta tese, como o uso de múltiplos dicionários e a compensação de movimentos com blocos sobrepostos e tamanho variáveis (OBMC). Em seguida, ensaios utilizando os realces propostos são comparados com seqüências sem realce, aplicados em diversos cenários. Por fim, comparamos com alguns métodos propostos na literatura [1, 2, 101], utilizando o mesmo teste descrito em [101], onde o 16^o quadro (sub-amostrado) deve ser super-resolvido a partir dos 1^o e 31^o quadros(-chave).

Na Figura 3.11, ilustramos o desempenho subjetivo da fusão de informação baseada nos pesos utilizando a Equação 3.8. Neste experimento, os quadros 1^o e 31^o da seqüência *Foreman* são utilizados como quadros-chave, o quadro em baixa-resolução a ser realçado é o 16^o. Ao utilizar apenas o 1^o quadro como exemplo, obtemos como resultado a Figura 3.11(a), cuja PSNR é 34,89 dB. Observe que alguns erros no processo de estimação de movimento induzem alguns erros no quadro realçado. Já a Figura 3.11(b) mostra o resultado do realce ao utilizar apenas o quadro 31^o como exemplo, o que resultou na PSNR de 35,80 dB. Contudo, ao utilizar os dois quadros como exemplo, e simplesmente escolher o bloco com menor distorção entre diferentes dicionários obtemos a Figura 3.11(c) com PSNR de 35,94 dB. Entretanto, o método proposto na Equação 3.8 permite a utilização dos quadros de referência de maneira mais eficiente, resultando na Figura 3.11(d), com a PSNR de 37,03 dB.

A Figura 3.12 mostra o desempenho da super-resolução proposta aplicado em diferentes tamanhos de GOPs. No geral, quanto maior a distância temporal menor a correlação entre os quadros. Ou seja, quanto maior o tamanho do GOP a tendência é que a PSNR da super-resolução seja menor. Entretanto, os movimentos dos conteúdos em cena, como por exemplo: movimentos não-translacionais dos objetos em cena, zoom, oclusões e etc, podem gerar alguns erros no processo de estimação de movimento, implicando em alguns erros no quadro realçado. Ou seja, alguns descasamentos na estimação de movimento implicam em incoerências entre a informação de alta-freqüência extraída do exemplo e a alta-freqüência do realce ideal, gerando as oscilações nas curvas das Figuras 3.12 e 3.13.

As curvas das Figuras 3.12 e 3.13 mostram o desempenho da super-resolução ao usar os seguintes dicionários:

1. dois dicionários combinados conforme a Seção 3.2.1 onde um deles contém informações do quadro anterior e o outro com posterior;

2. um dicionário contendo ambas informações do quadro anterior e posterior, portanto o casamento escolhido é o melhor dentre os exemplos do dicionário. Para efeitos de comparação, este tipo estimação birecional é utilizada em [101];

3. um dicionário contendo informações apenas do quadro anterior;

4. um dicionário contendo informações apenas do quadro posterior.

(a) 34,89 dB (b) 35,80 dB

(c) 35,94 dB (d) 37,03 dB

Figura 3.11: Ilustração do desempenho do realce utilizando combinações dos quadros exemplos. Os 1^o e 31^o são os quadros-chave de maior resolução utilizados como exemplo. A super-resolução do 16^o quadro da seqüência *Foreman* utilizando uma janela de busca de 64 $\times$ 64 pixels: (a) utilizando o 1^o quadro, (b) utilizando o 31^o quadro, (c) utilizando os blocos do quadro 1^o ou 31^o de melhor casamento (d) utilizando uma combinação linear entre os blocos dos dicionários compostos pelos quadros 1^o e 31^o.

Na Figura 3.12 e na Figura 3.13 a combinação de dois dicionários utilizados como exemplo geram, respectivamente, em média 1,30 dB e 1,06 dB superiores no processo de super-resolução se comparado ao uso de um único dicionário contendo as mesma informações.

Em seguida, comparamos os resultados da compensação de movimento com e sem sobreposição, ambos utilizando bloco de tamanhos variáveis. Os testes foram realizados com a seqüência *News* de tamanho CIF, onde o realce de resolução é feito no 16^o quadro e os quadros 1^o e 31^o são utilizados como exemplos. Ao utilizar a sobreposição na compensação de movimento (OBMC) obtemos um resultado de

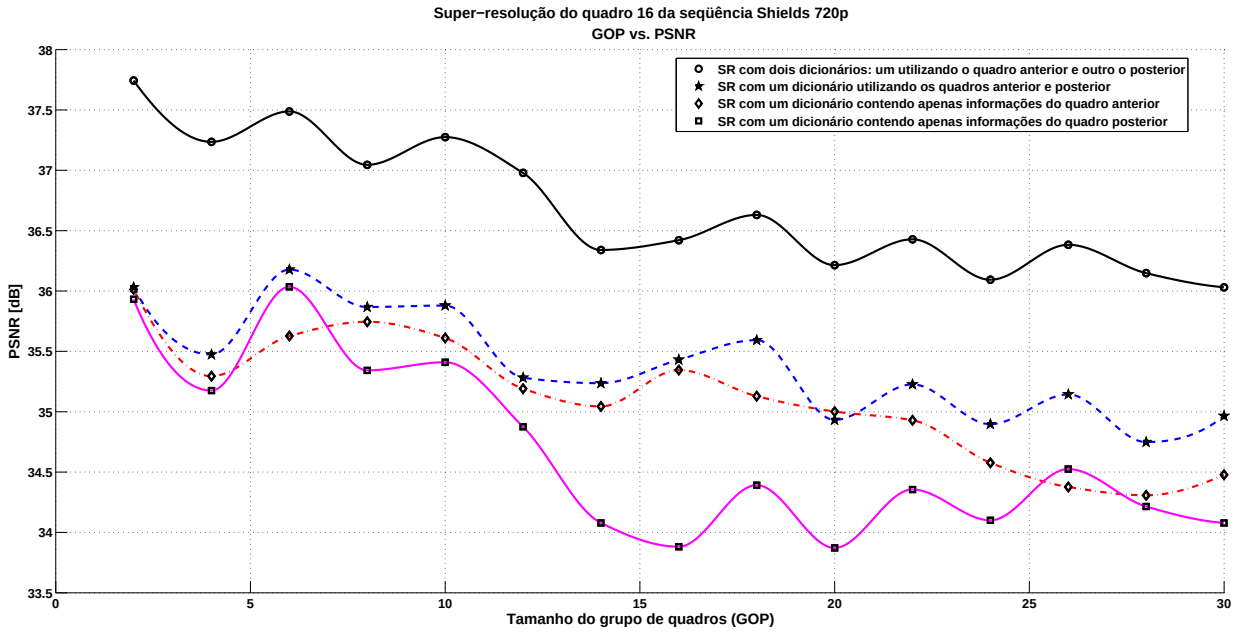


Figura 3.12: Resultados da super-resolução aplicado ao 16^o quadro da seqüência *Shields* utilizando diversos tamanhos de GOP. As curvas representam o desempenho da super-resolução ao usar diferentes dicionários: (1) Combinação de dois dicionários, um com informações do quadro anterior e o outro com posterior. Uso de um dicionário contendo (2) ambas informações do quadro anterior e posterior, (3) informações apenas do quadro anterior e (4) apenas do quadro posterior.

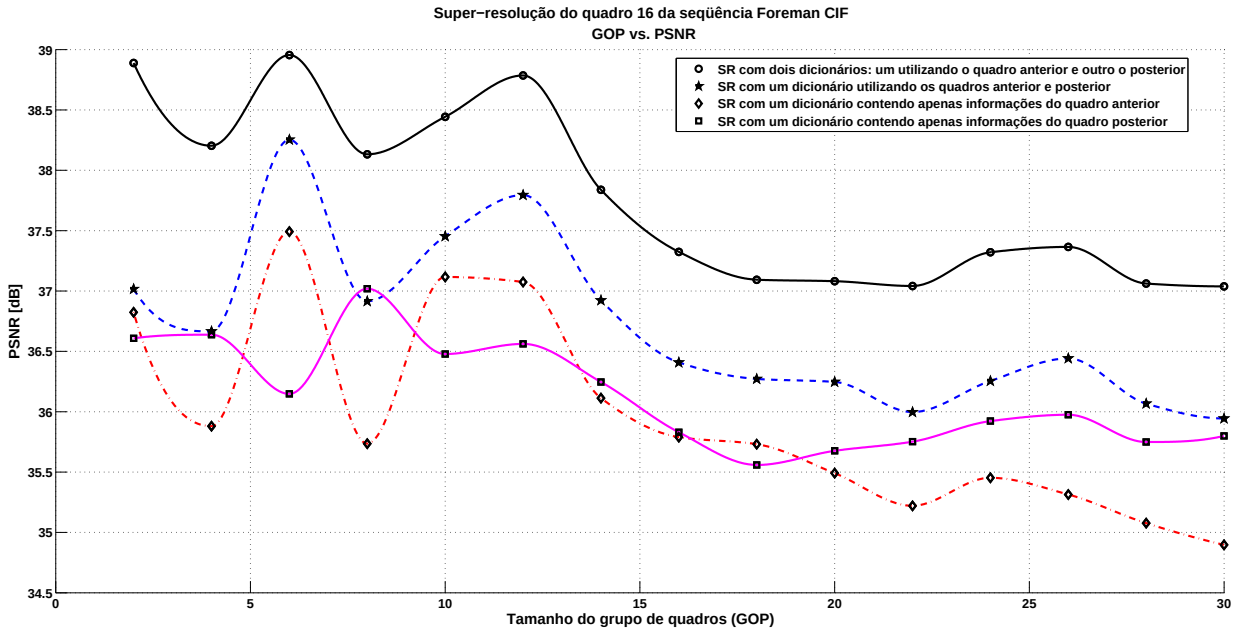
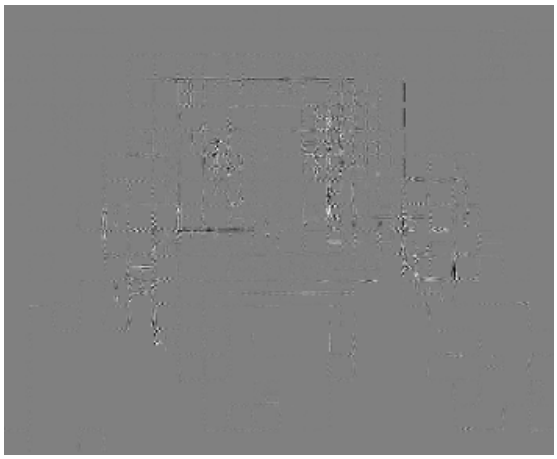


Figura 3.13: Resultados da super-resolução aplicado ao 16^o quadro da seqüência *Foreman* utilizando diversos tamanhos de GOP. As curvas representam o desempenho da super-resolução ao usar diferentes dicionários: (1) Combinação de dois dicionários, um com informações do quadro anterior e o outro com posterior. Uso de um dicionário contendo (2) ambas informações do quadro anterior e posterior, (3) informações apenas do quadro anterior e (4) apenas do quadro posterior.

38,81 dB após o processo de super-resolução. Ao trocar para compensação de movimento tradicional no processo de realce, a PSNR atingiu 38,50 dB. Os quadros podem ser observados nas Figuras 3.14(a) e 3.14(b), respectivamente. A Figura 3.14(c) ilustra as diferenças entre os dois processos de compensação de movimento utilizados nas super-resolução.

(a) 38,81 dB (b) 38,50 dB



(c)

Figura 3.14: Resultados da super-resolução aplicado ao 16^o quadro da seqüência *News*: utilizando compensação de movimento (a) com e (b) sem sobreposição. (c) Ilustração das diferenças entre (a) e (b).

Os testes foram realizados com 300 quadros das seqüências de vídeo: *Foreman*, *Mobile*, *Hall Monitor*, *Mother & Daughter* e *News* no formato CIF (352×288 pixels) e *Shields*, *Mobcal* e *Parkrun* em 720p (1280×720 pixels). As seqüências foram processadas para simular uma arquitetura de vídeo que utiliza resolução mista, utilizando os formatos QCIF (176×144 pixels) para os quadros-não-chave e CIF para

os quadros-chave. Os formatos de 360p (640×360 pixels) e 720p também foram utilizados na resolução mista, como mostra a Figura 3.15.

Os vídeos foram codificados utilizando o H.264 (JM 15.1) e o conjunto de parâmetros de quantização (QP) $\{22, 27, 32, 37\}$ foi utilizado, de forma a obter curvas de taxa-distorção [102]. A estimação de movimento utilizada no realce utilizou a janela de busca de 32×32 pixels para as seqüência no formato CIF e 64×64 pixels para os vídeos de alta definição.

Na Tabela 3.1, pode-se observar ganhos significativos do método de super-resolução proposto em comparação com os métodos de interpolação.

Tabela 3.1: Comparação objetiva [102] entre os vídeos interpolados com o filtro Lanczos e o método de super-resolução baseada em exemplos.

Seqüência	ganhos em PSNR
Foreman	2,47 dB
Mobile	2,28 dB
Mother and Daughter	1,23 dB
Shields	1,30 dB
Parkrun	1,66 dB
Mobcal	1,73 dB
Média	1,78 dB

Na Tabela 3.2 comparamos os resultados da super-resolução proposta utilizando o mesmo teste descrito em [101]. Os testes foram feitos sem compressão e o 16^o quadro (sub-amostrado) é super-resolvido utilizando o 1^o e 31^o quadros(-chave) como exemplo. Neste trabalho, um algoritmo de super-resolução híbrida (HSR) é apresentada, onde a super-resolução seleciona, utilizando uma limiarização, entre a super-resolução baseada na estimação de movimento (MSR) ou baseado em um dicionário treinado *on-the-fly* durante o processo de super-resolução.

O algoritmo MSR em [101] faz o uso de uma estimação de movimento hierárquica onde o cálculo da distância entre os blocos de baixa-resolução é feita combinando linearmente a distorção SAD e a taxa de codificação dos vetores de movimento. Além disso, a compensação de movimento bidirecional com sobreposição é feita escolhendo a informação do quadro anterior ou posterior.

No caso desta tese, a distância é calculada utilizando apenas a distorção SSD e a compensação de movimento bidirecional com sobreposição é feita utilizando blocos de tamanhos variáveis conforme

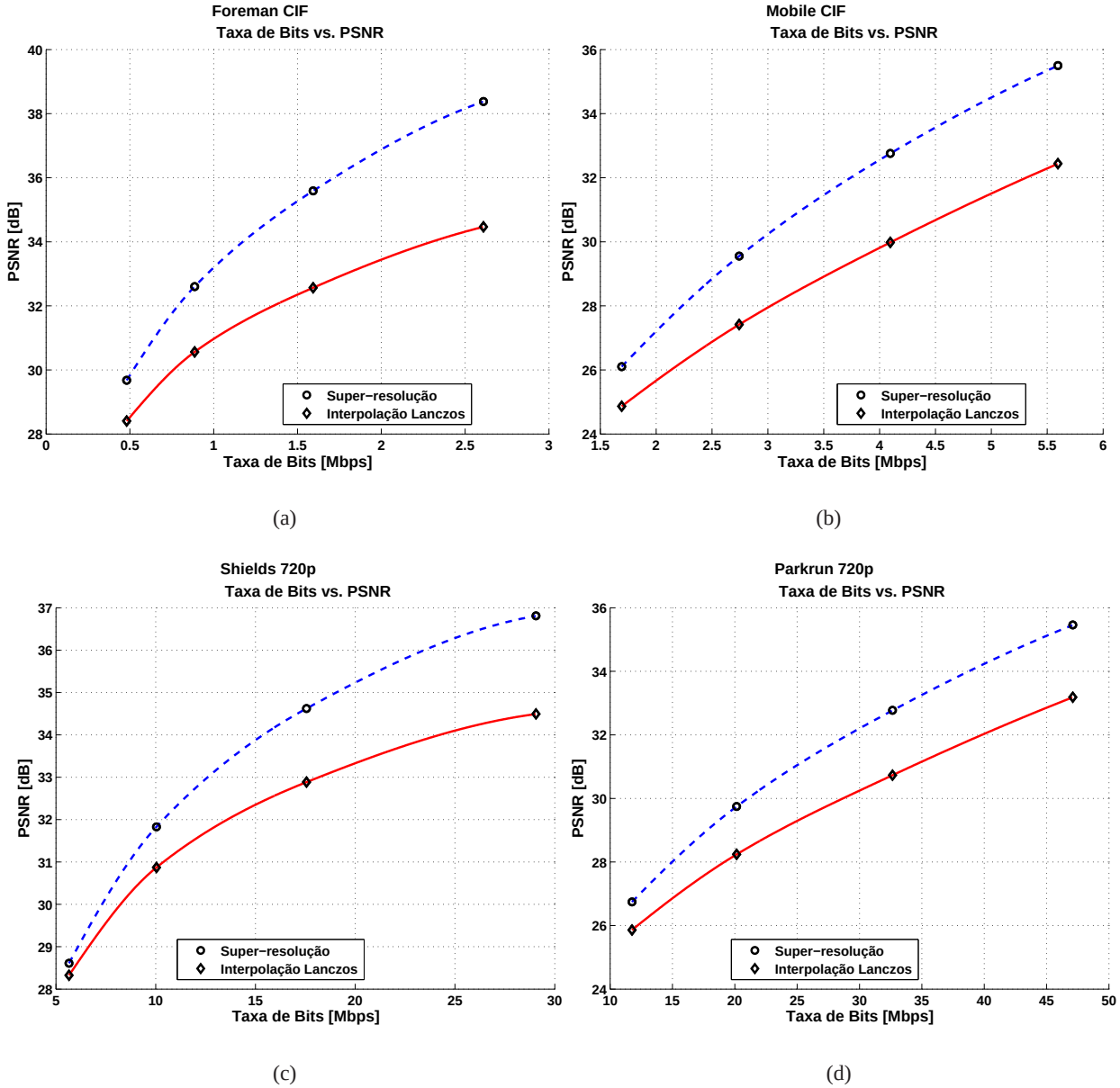


Figura 3.15: Comparação entre o vídeo interpolado e o método de super-resolução aplicado em seqüências de resolução mista: (a) *Foreman*, (b) *Mobile*, (c) *Shields* e (d) *Parkrun*.

apresentada na Seção 3.2.2. A informação utilizada para realçar a imagem é dada pela combinação linear entre as informações obtidas pela estimação bidirecional, cuja técnica foi descrita na Seção 3.2.1.

Tabela 3.2: Comparação em PSNR [dB] da super-resolução (SR) proposta na tese, SR por estimação de movimento (MSR) [101] e SR híbrido (HSR) [101].

Seqüência	Bicúbico [101]	Lanczos	SR em [1]	SR em [2]	MSR [101]	HSR [101]	SR proposta
Container	27,9 dB	27,4 dB	23,6 dB	30,7 dB	31,9 dB	33,2 dB	36,0 dB
Hall	29,1 dB	28,2 dB	24,2 dB	32,6 dB	37,4 dB	38,0 dB	41,1 dB
Mobile	22,9 dB	22,8 dB	20,4 dB	25,5 dB	24,5 dB	25,5 dB	27,1 dB
News	29,4 dB	30,1 dB	24,6 dB	34,1 dB	31,9 dB	36,1 dB	38,8 dB
Mobcal	27,7 dB	27,8 dB	24,2 dB	29,8 dB	30,9 dB	31,0 dB	35,0 dB
Shields	31,1 dB	33,1 dB	27,4 dB	34,9 dB	31,4 dB	32,7 dB	36,0 dB
Média	28,02 dB	27,92 dB	24,07 dB	31,27 dB	31,33 dB	32,75 dB	35,67 dB

A Figura 3.16 exemplifica os resultados subjetivos do quadro interpolado e o correspondente quadro após a super-resolução. A imagem original na Figura 3.16(a) é mostrada para que se possa avaliar subjetivamente o realce da qualidade nos quadros. O desempenho de diferentes algoritmos de interpolação também são comparados: a bicubica, mostrada na Figura 3.16(b) foi utilizada em [1, 2] e a interpolação Lanczos utilizada na super-resolução proposta é mostrada na Figura 3.16(c). A super-resolução proposta em [1], [2] e a técnica apresentada nesta tese são mostrados respectivamente nas Figuras 3.16(d), 3.16(e) e 3.16(f).

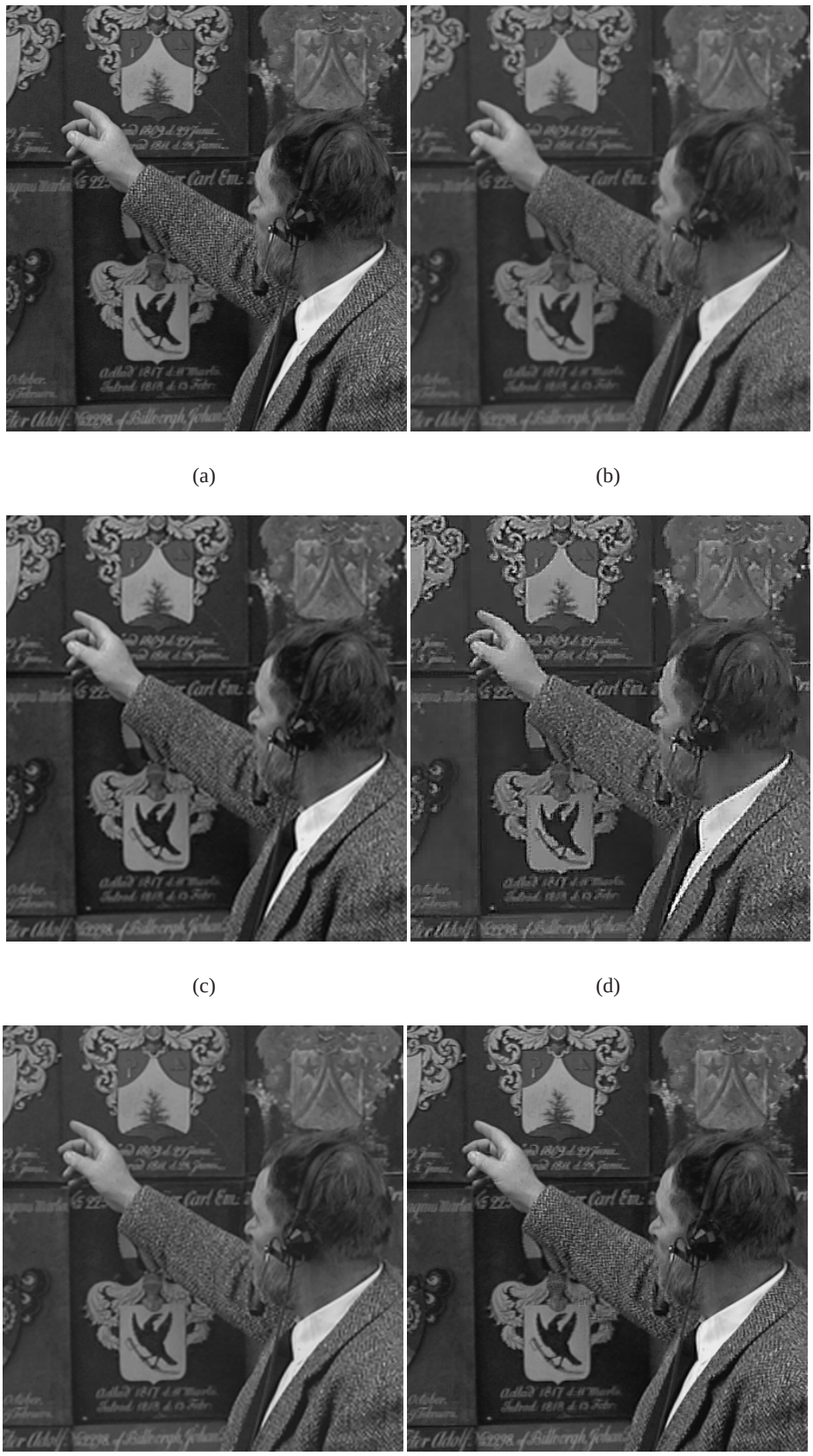
3.5 RESULTADOS EXPERIMENTAIS DA SUPER-RESOLUÇÃO DE VÍDEO EM BAIXA-RESOLUÇÃO UTILIZANDO FOTOGRAFIAS EM ALTA-RESOLUÇÃO COMO EXEMPLOS

Para simular o cenário onde fotografias são utilizadas para realçar um vídeo de baixa-resolução, utilizamos uma seqüência de vídeo com um quarto da resolução comprimido com H.264 e utilizamos um quadro redundante com o tamanho da resolução original a cada segundo, resultando em um GOP de 30, codificado com JPEG. As imagens comprimidas com JPEG utilizam uma matriz de quantização uniforme (Q). Como descrito nas seções anteriores, os métodos de super-resolução baseada em exemplos podem

ser aplicados em vários cenários. Os testes foram realizados com 300 quadros das seqüências de vídeo: *Foreman*, *Mobile*, *Hall Monitor*, *Mother & Daughter* e *News* no formato CIF (352×288 pixels) e *Shields*, *Mobcal* e *Parkrun* em 720p (1280×720 pixels). As curvas de taxa-distorção na Figura 3.17 comparam o desempenho da super-resolução utilizando fotografias, com o vídeo interpolado (sem o uso da fotografia) e ao vídeo com quadros que foram substituídos pelas fotografias redundantes. A super-resolução proposta permite um ganho significativo, se compararmos com a interpolação, o que indica que a técnica de realce permite a extrapolação da resolução de um vídeo. Ou seja, a super-resolução proposta aumenta a resolução da captura de um vídeo, cujo processo é fisicamente determinado pelo hardware (tamanho do CCD). Os ganhos objetivos podem podem ser verificados na Tabela 3.3.

Tabela 3.3: Comparação objetiva [102] entre o vídeo interpolado cujas fotografias de alta-resolução são substituídas pelos quadros de baixa-resolução e a super-resolução obtida utilizando fotografias como dicionário.

Seqüência	ganhos em PSNR
Foreman	1,97 dB
Shields	1,89 dB
Parkrun	0,93 dB
Mobcal	3,21 dB
Stockholm	0,71 dB
Média	1,74 dB

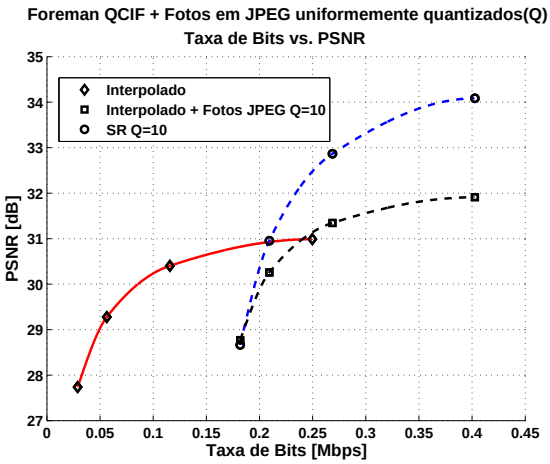


(a) (b)

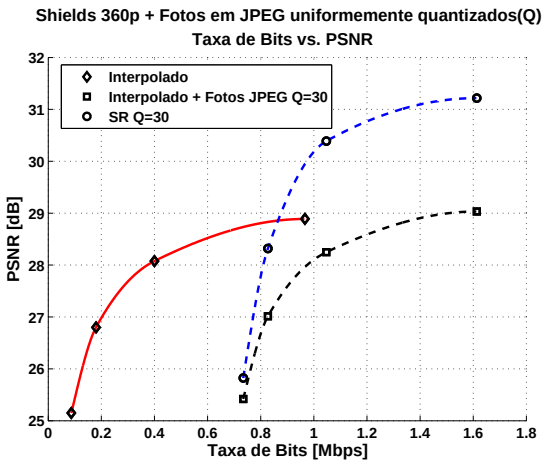
(c) (d)

(e) (f)

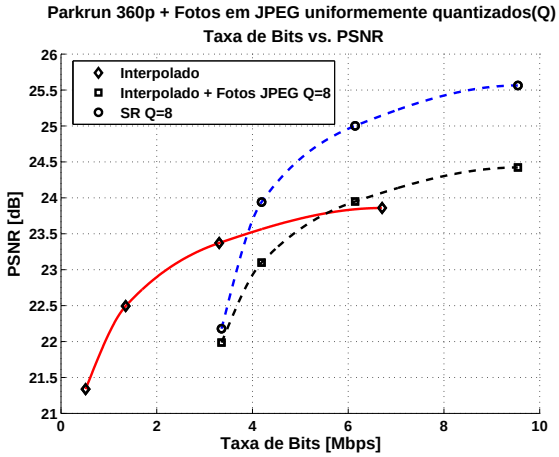
Figura 3.16: Região do 16^o quadro da seqüência *Shields*: (a)original, (b) interpolado com o filtro bicubico, (c) interpolado com o filtro Lanczos, (d) super-resolução pelo método [1], (e) super-resolução proposta em [2] e (f) super-resolução baseada em exemplos apresentada na tese.



(a)



(b)



(c)

Figura 3.17: Comparação entre o realce de vídeo com fotografias, o vídeo interpolado cujas fotografias de alta-resolução são substituídas pelos quadros de baixa-resolução e o vídeo interpolado aplicados às seguintes seqüências: (a) *Foreman*, (b) *Shields* e (c) *Parkrun*. As fotografias foram comprimidas com JPEG utilizando uma matriz de quantização uniforme (Q).

4 SUPER-RESOLUÇÃO NO DOMÍNIO TRANSFORMADO

4.1 INTRODUÇÃO

O método proposto de super-resolução por meio de transformada é baseado na DCT dos blocos de uma imagem, onde, por conseguinte, se determina os coeficientes de alta freqüência que serão adicionados às imagens de baixa resolução [7,103]. Contudo, o método proposto também permite o uso de outras técnicas de decomposição em freqüência, como por exemplo, as *wavelets*. A seguir, serão revistos os métodos de interpolação e decimação baseados na DCT, que embasam a super-resolução proposta.

Para decimar um bloco $\mathbf{b}$ contendo $m \times m$ pixels de uma imagem , para o tamanho $n \times n$, onde evidentemente $m > n$, deve-se utilizar:

$$T_{DCT}\{\mathbf{b}\} = \left[\begin{array}{c:c} \mathbf{B}_{00} & \mathbf{B}_{01} \\ \hdashline \mathbf{B}_{10} & \mathbf{B}_{11} \end{array}\right] \tag{4.1}$$

onde $T_{DCT}\{\mathbf{b}\}$ representa a transformada DCT de $\mathbf{b}$. Os coeficientes da transformada são expressados como uma matriz particionada, sendo $\mathbf{B}_{00}$ uma sub-matriz $(n \times n)$ que contém os coeficientes de baixa freqüência. $\mathbf{B}_{01}$, $\mathbf{B}_{10}$ e $\mathbf{B_{11}}$ são matrizes de tamanhos: $(m-n) \times n$, $n \times (m-n)$ e $(m-n) \times (m-n)$, respectivamente, que contêm os coeficientes de alta freqüência.

A decimação do bloco $\mathbf{b}$ da imagem é obtida após calcularmos a DCT inversa da sub-matriz de baixa-freqüência $\mathbf{B}_{00}$, descartando assim os componentes de alta freqüência [104]:

$$\mathbf{b}_{dsz} = T_{IDCT}\{s_{dsz}[\mathbf{B}_{00}]\}. \tag{4.2}$$

onde, por conta das diferenças de tamanho entre as DCTs inversa e direta, a sub-matriz de baixa-freqüência $\mathbf{B}_{00}$ deve ser multiplicada por um fator de escala $s_{dsz} = n/m$ seguida da transformada inversa $T_{IDCT}\{\cdot\}$ para obtermos o bloco decimado $\mathbf{b}_{dsz}$ de tamanho $n \times n$.

A interpolação de um bloco $n \times n$ da imagem, redimensionado para $m \times m$, pode ser obtida ao inserir coeficientes de amplitude nula de alta freqüência e em seguida calcular a DCT inversa [104]. Por exemplo, a interpolação de $\mathbf{b}_{dsz}$ é obtida ao assumirmos que os valores das sub-matrizes $\mathbf{B}_{01}$, $\mathbf{B}_{10}$ e $\mathbf{B}_{11}$, da Equação 4.1 são nulos formando assim um bloco de $m \times m$ pixels dados por

$$\mathbf{b}_{usz} = T_{IDCT}\left\{\left[\begin{array}{c:c} \mathbf{B}_{00} & \mathbf{0} \\ \hdashline \mathbf{0} & \mathbf{0} \end{array}\right]\right\}. \tag{4.3}$$

Em outras palavras, a interpolação utilizando DCT é obtida ao dividirmos a imagem em blocos de tamanho $n \times n$, de forma a determinar os coeficientes de cada sub-matriz. A sub-matriz de baixa freqüência $\mathbf{B}_{00}$ é associada ao bloco transformado de baixa resolução $T_{IDCT}\{\mathbf{b}_{dsz}\}$, em seguida adicionamos os coeficientes nulos de alta freqüência e aplicamos a DCT inversa, conforme descrito na Equação 4.3. Apesar da interpolação por DCT resultar em uma qualidade objetiva maior que as interpolação no domínio espacial, como a bilinear e a bicubica, sua qualidade subjetiva não é satisfatória, pois a adição de coeficientes de amplitude nula descrita pela Equação 4.3 usualmente gera efeitos de blocos [105].

O método de super-resolução proposto tem por objetivo melhorar a qualidade da interpolação, utilizando-se da estimação dos coeficientes de alta freqüência baseados em imagens ou quadros com resolução maiores, ao invés de assumir que as sub-matrizes de alta freqüência sejam nulas. Portanto, as imagens de alta resolução servirão como fonte de coeficientes de alta freqüência que são utilizados para preencher os coeficientes de detalhe para aplicar a super-resolução nas imagens de baixa-resolução. Para cada bloco $\widehat{\mathbf{b}}$ da imagem-exemplo de alta resolução, os coeficientes são descritos como:

$$T_{DCT}\left\{\widehat{\mathbf{b}}\right\} = \left[\begin{array}{c:c} \widehat{\mathbf{B}}_{00} & \widehat{\mathbf{B}}_{01} \\ \hdashline \widehat{\mathbf{B}}_{10} & \widehat{\mathbf{B}}_{11} \end{array}\right], \tag{4.4}$$

onde as sub-matrizes de alta freqüência $\widehat{\mathbf{B}}_{01}$, $\widehat{\mathbf{B}}_{10}$ e $\widehat{\mathbf{B}}_{\mathbf{11}}$ são utilizados para completar a informação inexistente de alta freqüência em blocos co-localizados $\mathbf{b}_{usz}$ da imagem interpolada, que originalmente eram de baixa-resolução. Portanto, o bloco da imagem que sofreu o processo de super-resolução $\mathbf{b}_{SR}$ é dado por

$$\mathbf{b}_{SR} = T_{IDCT}\left\{\left[\begin{array}{c:c} \mathbf{B}_{00} & \widehat{\mathbf{B}}_{01} \\ \hdashline \widehat{\mathbf{B}}_{10} & \widehat{\mathbf{B}}_{11} \end{array}\right]\right\} \tag{4.5}$$

onde a sub-matriz $\mathbf{B}_{00}$ são coeficientes DCT de baixa freqüência remanescentes do processo de interpolação, como descrito na Equação 4.3. Ao aplicar a super-resolução em cada bloco de baixa-resolução, obtemos como resultado uma imagem com alta-resolução cujos coeficientes de alta freqüência são oriundos das imagens de maior resolução.

4.2 SUPER-RESOLUÇÃO NO DOMÍNIO DA DCT EM VÍDEOS COMPOSTOS POR QUADROS DE RESOLUÇÃO MISTA

Aplicamos a técnica de super-resolução no domínio da DCT em vídeos compostos por quadros de resolução mista. A Figura 4.1 mostra o diagrama de blocos para o problema em questão. Primeiramente, os quadros-chave e não-chave são separados. Em seguida, os quadros-chave são sub-amostrados e interpolados utilizando as Equações 4.1 e 4.3, gerando assim uma versão interpolada do quadro-chave com qualidade similar ao dos quadros-não-chave. Em seguida, fazemos uma estimação de movimento bidirecional, em blocos de tamanho variável [97], entre as versões interpoladas dos quadros-chave e os quadros-não-chave, o que resulta em vetores de movimento na escala sub-pixel. Os vetores de movimento calculados nestes processos irão compensar os blocos utilizando a sobreposição supracitada [14] utilizando como referência os quadros-chave. Após obter dois quadros-chave compensados oriundos da estimação/compensação de movimento bidirecional, devem-se fundir os mesmos utilizando a Equação 3.9 para obter uma boa estimação do quadro de alta-resolução. Por fim, a super-resolução é realizada ao interpolar os quadros-não-chave e substituir as informações de alta freqüência do quadro estimado, conforme as Equações 4.3, 4.4 e 4.5.

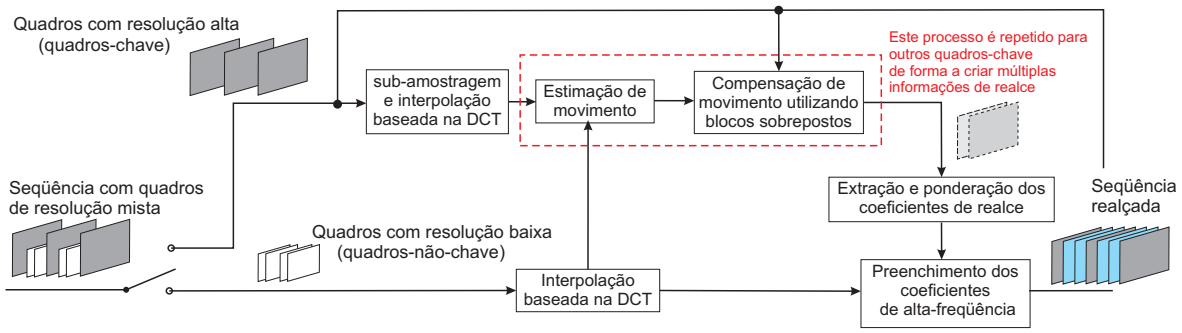


Figura 4.1: Super-resolução no domínio DCT aplicado a vídeo contendo quadros de resolução mista.

4.3 SUPER-RESOLUÇÃO NO DOMÍNIO DA DCT PARA IMAGENS DE MÚLTIPLAS VISTAS E RESOLUÇÃO MISTA

Além do seu uso para redução de complexidade em codificação de vídeo [17, 95], as arquiteturas de resolução mista também são empregadas na redução de tamanho de arquivo em vídeo estreoscópico [106, 107]. Nesse caso, ao invés de intercalar temporalmente quadros de alta e baixa-resolução, uma vista de baixa-resolução é apresentada ao olho esquerdo, por exemplo, enquanto a vista de alta-resolução é

reservada ao olho direito. Métodos de super-resolução tem sido empregados no processamento de vídeo em resolução mista para aliviar o problema de cintilamento (*flickering*) durante a visualização. Porém, o vídeo estereoscópico em resolução mista (também chamado de assimétrico) é geralmente visualizado em resoluções diferenciadas e sem processamento. Este fenômeno é justificado em estudos psico-visuais [108, 109] que indicam que a agudeza e a percepção de profundidade da imagem estereoscópica são determinadas pelo canal de alta-resolução. No entanto, em casos mais genéricos como os sistemas de múltiplas vistas, a resolução mista pode não ser diretamente aplicável. As diferenças de qualidade entre as diferentes vistas podem ser prejudiciais a algumas das aplicações almejadas pelos sistemas de múltiplas vistas. Por exemplo, devido à sua natureza monoscópica, a navegação entre vistas dentro de um vídeo com ponto de vista livre (*free view-point video*) apresentará significativas diferenças em qualidade entre as vistas de alta e baixa-resolução.

Para superar tais limitações, um método de super-resolução para uso em arquiteturas de múltiplas vistas em resolução mista foi proposto [13]. A arquitetura está ilustrada na Figura 4.2 e consiste em múltiplas seqüências de vídeo de diferentes pontos de vista e resoluções e seus mapas de profundidade correspondentes. Os mapas de profundidade [110] foram mantidos na resolução máxima pois são eficientemente codificáveis e representam uma pequena porcentagem do tamanho total de dados [10]. Os mapas são utilizados para estabelecer as correspondências entre vistas.

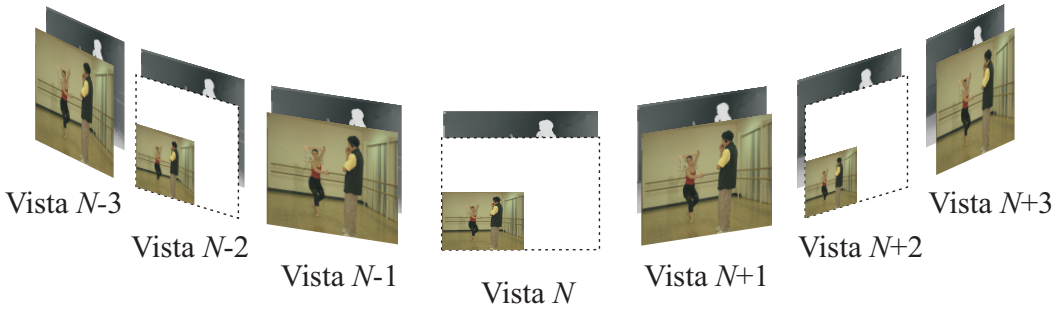


Figura 4.2: A arquitetura de múltiplas vistas em resolução mista com mapas de profundidade.

O método de super-resolução de [13] baseia-se no uso de filtros lineares para interpolar as imagens de baixa-resolução e para isolar o conteúdo de alta freqüência em imagens vizinhas de alta-resolução. Ambas operações foram implementadas no domínio espacial, porém ambas apresentam alternativas atraentes no domínio da transformada. Ganhos objetivos de qualidade obtidos através de operações de interpolação no domínio da transformada relativos à interpolação linear de parâmetros fixos, tais como bilinear, são apresentados em [104]. A qualidade visual de resultados pode ser melhorada ao combinar interpolação

baseada em DCT, responsável por preservar os coeficientes de baixa freqüência, com uma estimação baseada em filtros Wiener dos coeficientes de alta freqüências [105]. Além de bons resultados de interpolação, o uso de transformadas constitui um domínio natural para a decomposição e isolamento de freqüências em imagens.

Esta tese apresenta um método de super-resolução para uso em arquiteturas de múltiplas vistas em resolução mista com informação de profundidade. O conteúdo de alta freqüência presente nas imagens adjacentes de alta-resolução é usado para realçar as imagens de baixa-resolução, onde introduzimos as operações no domínio da transformada para realizar a interpolação e o isolamento dos coeficientes de alta freqüência. O método proposto preserva coeficientes DCT de baixa freqüência presentes na imagem de baixa-resolução e complementa os mesmos com coeficientes DCT de alta freqüência provenientes de vistas de alta-resolução adequadamente projetadas. A projeção é realizada com uso de informação de profundidade.

O método proposto de realce no domínio da transformada é ilustrado na Figura 4.3. As imagens de alta-resolução disponíveis na arquitetura de resolução mista, que utiliza vídeos com o seguinte formato: múltiplas vistas e mapa de profundidade, são inicialmente projetadas em um ponto de vista correspondente à uma imagem de baixa-resolução como indica o bloco de Projeção de Vista da Figura 4.3. O bloco de Interpolação é responsável por aumentar o tamanho da imagem de baixa-resolução em tamanhos compatíveis aos de alta-resolução. Finalmente, a super-resolução baseada em blocos DCT adiciona os coeficientes de alta-freqüência da vista projetada na imagem de baixa-resolução.

4.3.1 Projeção de Vista

A projeção de vista com boa qualidade é um componente essencial para a super-resolução proposta. A técnica de projeção utilizada na arquitetura proposta faz a renderização baseada no mapa de profundidade e possui como entradas a imagem de alta-resolução V_{N-1}, mapas de profundidade D_{N-1} and D_N e possui como saída a síntese de uma imagem na N-ésima vista $\widehat{V_N}$. O conhecimento dos parâmetros intrínsecos das câmeras $\mathbf{A}$, como por exemplo o centro óptico da câmera, a distância focal e relação pixel/milímetro[2] do sensor, etc. Já a matriz de rotação $\mathbf{R}$, vetor de translação $\mathbf{t}$ são utilizados para projetar a localização $(\widehat{u}, \widehat{v})$ do pixel na câmera N nas coordenadas absolutas tridimensionais(x, y, z). Maiores detalhes podem ser obtidos nas referências [8, 111]:

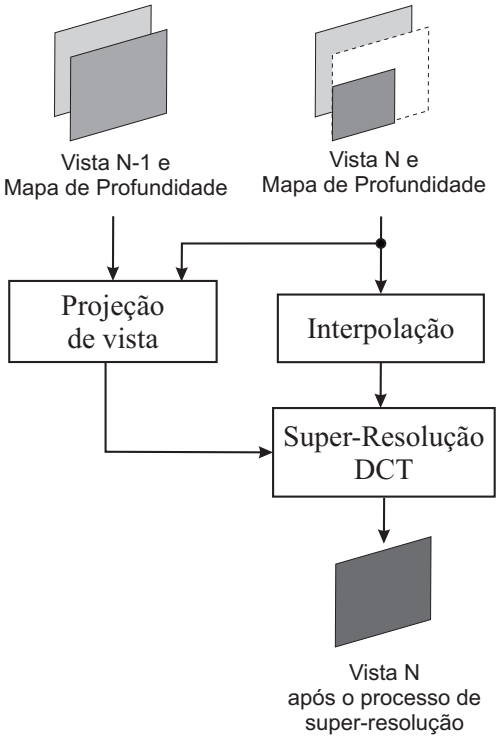


Figura 4.3: Diagrama de blocos do método proposto de super-resolução baseado em transformadas.

$$(x, y, z)^T = \mathbf{R}_N {\mathbf{A}_N}^{-1} (\widehat{u}, \widehat{v}, 1)^T D_N(\widehat{u}, \widehat{v}) + \mathbf{t}_N. \tag{4.6}$$

Em seguida, as coordenadas absolutas são re-projetadas na câmera $N-1$ na posição (u, v):

$$(uw, vw, w)^T = \mathbf{A}_{N-1} {\mathbf{R}_{N-1}}^{-1} [(x, y, z)^T - \mathbf{t}_{N-1}]. \tag{4.7}$$

Geralmente, nem todos os pixels possuem correspondências entre as vistas. Um teste de consistência também é aplicado de forma a identificar possíveis erros de correspondência. As coordenadas (u, v) são arredondadas para os valores inteiros mais próximos e projetadas de volta à câmera N. Porém, se a distância Euclidiana entre as posições resultantes desta última projeção e as coordenadas originais $(\widehat{u}, \widehat{v})$ for menor que um limiar especificado (tipicamente 1.0) a correspondência é aceita, caso contrário ela é rejeitada. A projeção da vista é completada ao substituir $(\widehat{u}, \widehat{v})$ pela interpolação bilinear da amostra validada na posição correspondente (u, v) na vista V_{N-1}. As posições onde o teste de correspondência não logrou êxito permanecem na imagens como buracos, exemplificados pela Figura 4.4. Observe que no método proposto os buracos oriundos da projeção são preenchidos por pixels da imagem interpolada (originalmente com baixa-resolução), portanto, não há componentes de alta freqüência para essas regiões

(a) (b)

Figura 4.4: (a) Seqüência *Ballet* (vista 0, imagem 0) e (b) sua projeção para vista 1 (buracos mostrados em branco).

que contribuam ao processo de super-resolução. A extensão do método de super-resolução para uso com duas ou mais vistas adjacentes poderá diminuir a área correspondente aos buracos de projeção [13].

4.4 SUPER-RESOLUÇÃO POR TRANSFORMADA APLICADAS A VÍDEOS COM RESOLUÇÃO MISTA

Nesta seção iremos avaliar o desempenho da super-resolução no domínio da transformado (TDSR, do inglês, *transform-domain super-resolution*), que será comparado com a super-resolução no domínio dos pixels (PDSR, do inglês, *pixel-domain super-resolution*) apresentado no Capítulo 3. Os testes foram realizados com várias seqüências de resolução mista de tamanho CIF (352×288 pixels) e 720p (1280×720 pixels) para os quadros-chave e QCIF (176×144 pixels) e 360p (640×360 pixels) para os quadros-não-chave, respectivamente, e GOP igual a 2, codificados com H.264 (JM 15.1) utilizando QPs iguais a $\{22, 27, 32, 37\}$ para o levantamento das curvas de taxa distorçao [102]. Na super-resolução a estimação de movimento utilizada foi a busca completa (*full search*) em uma janela de 32×32 e dois quadros de referência (os quadros-chave anterior e o posterior ao quadro-não-chave em processo de realce).

As curvas da Figura 4.5 mostram um ganho de desempenho significativo do método proposto, principalmente em menores taxas. Na Tabela calculamos as diferenças de PSNR entre o TDSR e o PDSR que utiliza a sub-amostragem e a interpolação pela DCT. No caso da seqüência *Foreman* a diferença de PSNR entre as curvas é de 0,194 dB, mas no ponto de maior qualidade a diferença chega a ser maior que

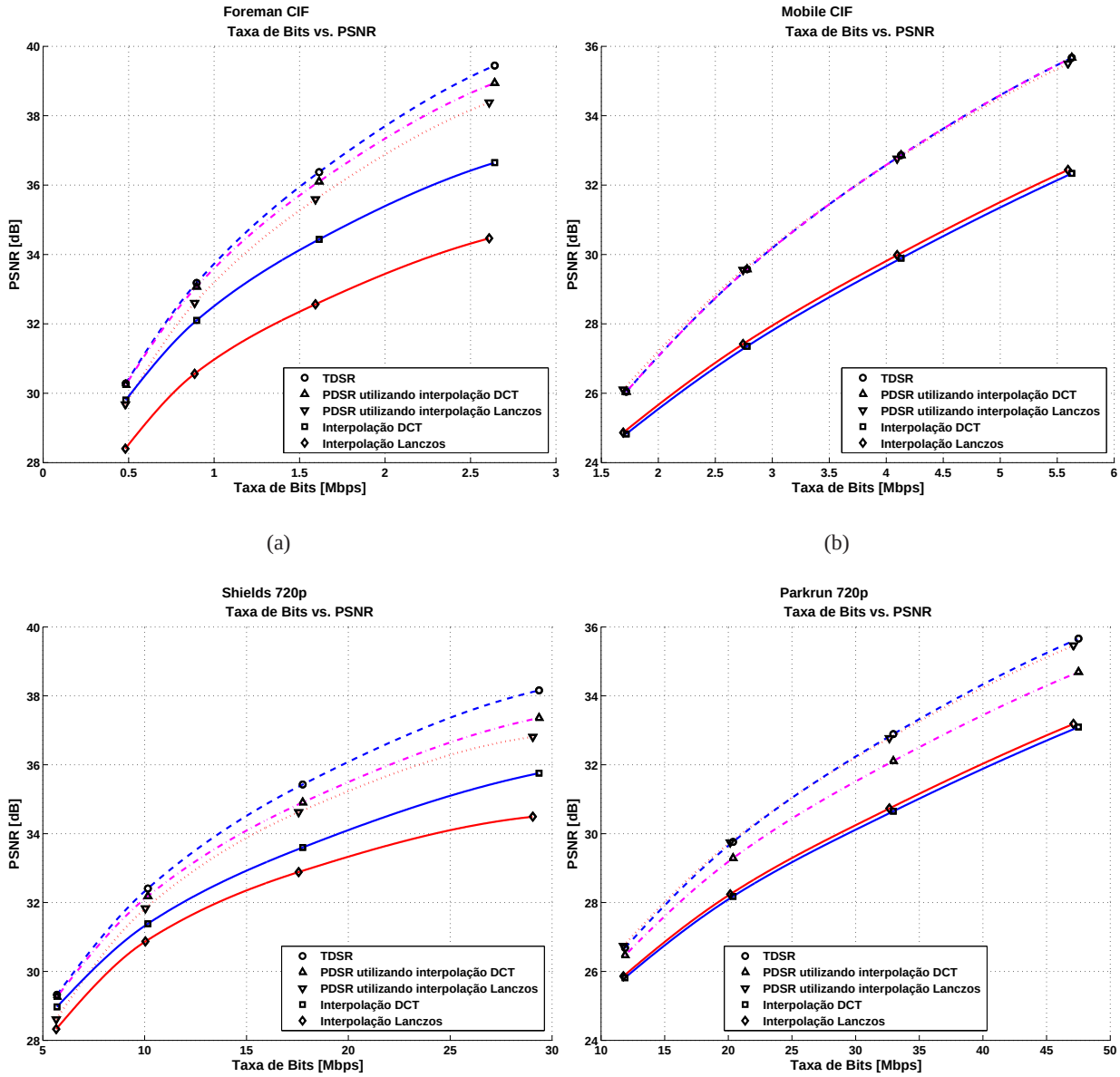


Figura 4.5: Comparação entre o método de super-resolução no domínio da transformada e no domínio dos pixels utilizando as interpolações Lanczos e DCT aplicados as seqüências: (a) *Foreman* e (b) *Mobile* cujos quadros-chave são de tamanho CIF e os quadros-não-chave de tamanho QCIF; (c) *Shields* e (d) *Parkrun* têm quadros-chave de tamanho 720p e quadros-não-chave de tamanho 360p.

0,35 dB. Entretanto, quando comparamos os métodos anteriores com o PDSR utilizando o filtro Lanczos, os resultados objetivos são superiores na maioria dos casos.

Tabela 4.1: Diferença de PSNRs [102] entre os métodos: TDSR-DCT, PDSR-DCT, PDSR-Lanczos e interpolação por DCT, e a interpolação Lanczos.

	Método vs.	
Seqüência	Interp. com filtro Lanczos	Diferença de PSNR
Foreman (CIF)	TDSR-DCT	3,1168 dB
	PDSR-DCT	2,9225 dB
	PDSR-Lanczos	2,4748 dB
	Interpolação por DCT	1,6530 dB
Shields (720p)	TDSR-DCT	2,0111 dB
	PDSR-DCT	1,6419 dB
	PDSR-Lanczos	1,3002 dB
	Interpolação por DCT	0,6432 dB
Parkrun (720p)	TDSR-DCT	1,6502 dB
	PDSR-DCT	1,0741 dB
	PDSR-Lanczos	1,6551 dB
	Interpolação por DCT	-0,1209 dB

Os resultados sujetivos da seqüência *Foreman*, codificado com o H.264 usando $QP = 22$, são mostrados na Figura 4.6. Na Figura 4.6(a) um quadro de alta-resolução codificado é mostrado como resultado ideal do processo de super-resolução, na Figura 4.6(b) o resultado da interpolação a partir de um quadro codificado em resolução baixa e na Fig. 4.6(c) a PDSR do caso anterior. Em seguida, a Figura 4.6(d) mostra o resultado da interpolação utilizando a DCT que um quadro codificado em resolução baixa e na Figura 4.6(e) o resultado da TDSR proposta. Podemos observar o método de TDSR pode ser subjetivamente melhor que os métodos de PDSR em regiões mais suaves e com texturas. Entretanto, apesar dos ganhos objetivos, artefatos de *ringing* originados do método de interpolação via DCT podem não ser completamente removidos após o método de TDSR.

(a) (b) (c)

(d) (e)

Figura 4.6: Detalhes da seqüência *Foreman*: (a) quadro com resolução alta codificado com H.264, (b) interpolação utilizando Lanczos de um quadro de resolução baixa codificado com H.264, (c) PDSR do caso anterior, (d) interpolação utilizando DCT de um quadro de baixa-resolução codificado com H.264 e (e) TDSR do caso anterior.

4.5 RESULTADOS EXPERIMENTAIS DA SUPER-RESOLUÇÃO POR TRANSFORMADA APLICADAS EM IMAGENS COM MÚLTIPLAS VISTAS DE MAPAS DE PROFUNDIDADE

O desempenho do método proposto foi avaliado com um conjunto publicamente disponível de imagens reais e sintéticas. Devido à falta de conteúdo de alta freqüência, as imagens reais *Ballet* e *Breakdancers* foram redimensionadas para 512×384 e 256×192 *pixels*, respectivamente, antes de serem testadas. As imagens em múltiplas vistas, disponibilizadas em conjunto com os mapas de profundidade ou de disparidade, foram reduzidas de forma a comporem a arquitetura de resolução mista apresentada na Figura 4.2. O método de SR é aplicado a cada imagem em baixa-resolução, baseado na vista mais próxima em alta resolução. As operações utilizam a DCT do tipo II em todos os casos, com blocos de tamanho original

8×8 e tamanho reduzido 4×4 ($m = 8$ e $n = 4$), o que resulta em um fator de redução de 2 em cada direção.

Os primeiros testes comparam os resultados de interpolação da imagem em baixa resolução utilizando um filtro linear de alto desempenho (com um *kernel* Lanczos) e com o método baseado em DCT apresentado no Capítulo 4. A Tabela 4.2 mostra uma ligeira piora do método baseado em DCT em relação à interpolação com *kernel* Lanczos, de -0,32 dB, em média. Note que o método baseado em DCT simplesmente acrescenta coeficientes de amplitude nula como uma estimativa dos componentes de alta freqüência.

Tabela 4.2: Comparação em PSNR de métodos de interpolação.

Sequência	*Kernel* Lanczos	DCT	Ganho em PSNR
Ballet	34,01 dB	33,71 dB	-0,30 dB
Breakdancers	35,47dB	34,95 dB	-0,52 dB
Barn1	27,76 dB	27,49 dB	-0,27 dB
Barn2	31,06 dB	30,74 dB	-0,32 dB
Bull	32,46 dB	32,23 dB	-0,23 dB
Map	28,00 dB	27,56 dB	-0,44 dB
Poster	26,46 dB	26,06 dB	-0,40 dB
Sawtooth	28,32 dB	27,93 dB	-0,39 dB
Venus	28,63 dB	28,40 dB	-0,23 dB

O segundo conjunto de testes compara o método proposto de SR no domínio da transformada com um método de SR no domínio espacial. Este segundo método utiliza uma interpolação linear baseada no *kernel* Lanczos e a extração espacial de alta freqüência assim como descrito em [13]. Porém, para efeito de comparação, a projeção da vista adjacente é idêntica em ambos os métodos (nos domínios espacial e da transformada). A Tabela 4.3 indica que o método no domínio da transformada é superior ao método no domínio espacial em todas as sequências, exceto por uma. Verifica-se um ganho médio de 0,16 dB, chegando a 0,5 dB para a sequência *Bull*. Observe que o método no domínio da transformada utiliza uma interpolação de desempenho objetivo inferior, como indicado na Tabela 4.2, mas que se mostra mais adequado no processo de SR.

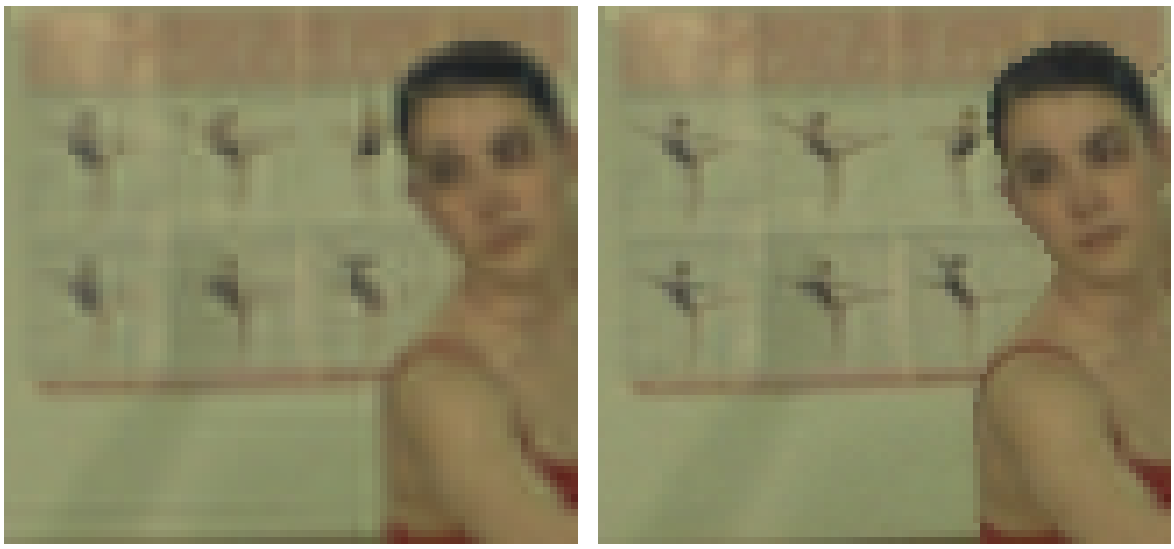
Uma avaliação subjetiva do método proposto pode ser realizada por meio das imagens da Figura 4.7. Para a sequência *Ballet*, a SR da vista 1 baseada em DCT usando uma imagem de alta-resolução correspondente à vista 2 pode ser comparada com a interpolação da vista 1 baseada em DCT. Detalhes de

Tabela 4.3: Comparação em PSNR de métodos de super resolução nos domínios espacial e da transformada.

Sequência	*Kernel* Lanczos	DCT	Ganho em PSNR
Ballet	36,18 dB	**36,31 dB**	0,15 dB
Breakdancers	38,69 dB	**38,84 dB**	0,15 dB
Barn1	35,83 dB	**36,22 dB**	0,39 dB
Barn2	38,40 dB	**38,50 dB**	0,10 dB
Bull	37,96 dB	**38,46 dB**	0,50 dB
Map	31,20 dB	**31,24 dB**	0,04 dB
Poster	33,93 dB	**34,09 dB**	0,16 dB
Sawtooth	**33,72** dB	33,32 dB	-0,40 dB
Venus	35,61 dB	**35,99** dB	0,38 dB

alta freqüência foram inseridos pelo método de SR, ressaltando o contorno na face da bailarina e na textura do fundo da imagem. Note que nesse experimento projetou-se a vista 2 e não a vista 0 como ilustrado no exemplo da Figura 4.4. Estas melhoras refletem o ganho atingido em termos de PSNR, de 2,60 dB, que pode ser obtido comparando a segunda coluna das Tabelas 4.2 e 4.3.

Para a sequência sintética *Barn1*, a diferença de PSNR entre a SR baseada em DCT e a interpolação baseada em DCT é de 8,73 dB. A Figura 4.8 permite uma comparação subjetiva. Observe que a inserção de componentes de alta freqüência pelo método proposto resulta em uma imagem mais detalhada e definida.



(a) (b)

Figura 4.7: Detalhe parcial da vista 1 da seqüência *Ballet*: (a) Imagem interpolada com DCT (33,71 dB) e (b) Imagem submetida à SR baseada na DCT (36,31 dB).

(a)

(b)

Figura 4.8: Vista 1 da seqüência *Barn1*: (a) Imagem interpolada com DCT (27,49 dB) e (b) Imagem submetida à SR baseada na DCT (36,22 dB).

5 GENERALIZAÇÃO DO REALCE BASEADO EM EXEMPLOS

5.1 INTRODUÇÃO

Nesta tese propõe-se a generalização do método de super-resolução descrito no Capítulo 3 para realçar a qualidade das imagens, onde a variação da qualidade é dada pelo quantizador, por uma filtragem ou por algum outro processo de degradação. De maneira análoga à super-resolução baseada em exemplos, são utilizados dicionários contendo porções de imagens de referência com alta qualidade $\dot{f}_{HQ}$ associadas às suas versões de baixa qualidade $\dot{f}_{LQ}$. E dada uma porção da imagem a ser realçada $\dot{\mathbf{g}}_{\mathbf{LQ}_k}$, uma busca em $\dot{f}_{LQ}$ é feita até que se encontre um bom par $\dot{\mathbf{f}}_{\mathbf{LQ}_\nu}$, permitindo que a informação de realce $\dot{\mathbf{f}}_{\mathbf{HQ}_\nu} - \dot{\mathbf{f}}_{\mathbf{LQ}_\nu}$ seja adicionada à $\dot{\mathbf{g}}_{\mathbf{LQ}_k}$.

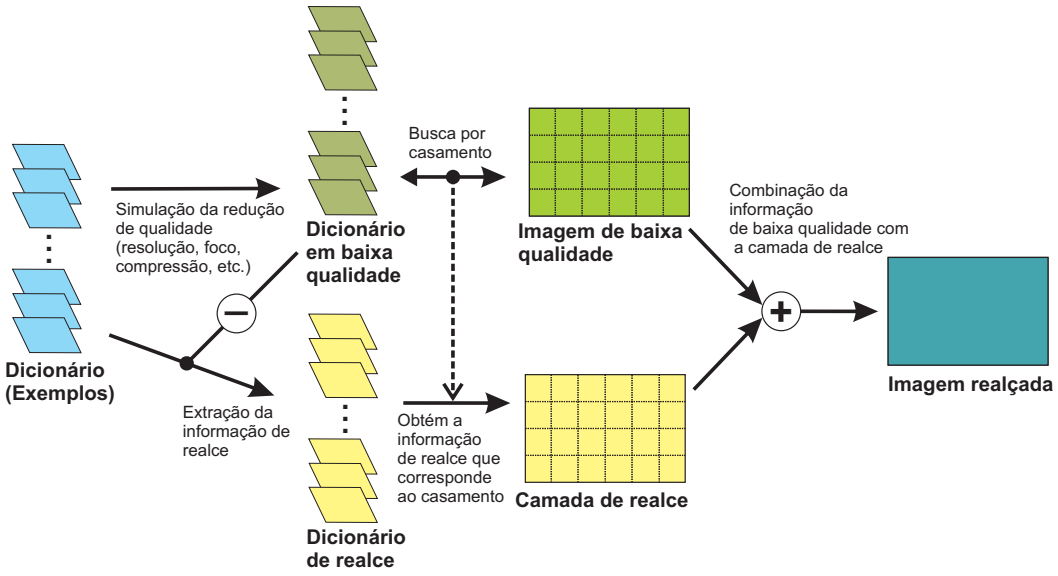


Figura 5.1: Diagrama geral do realce baseado em exemplos.

A Figura 5.1 mostra um dicionário utilizado como exemplo, cujo conteúdo é separado em dois conjuntos: o de qualidade reduzida e sua respectiva informação de realce. Em seguida é feita uma busca no conjunto de blocos com qualidade reduzida para selecionar o melhor casamento com o bloco que se quer realçar. A informação de realce utilizada corresponde ao casamento escolhido. Ao realizar este procedimento para todos os blocos de uma imagem, uma camada de realce é estimada. Por fim, a imagem realçada é obtida pela combinação da imagem com qualidade reduzida e a camada de realce.

Neste caso, o processo de realce baseado em exemplos é uma generalização da super-resolução baseada em exemplos, pois a última trata especificamente de aumentar a resolução de imagens. Entretanto, o processo de realce baseado em exemplos não deve aumentar a qualidade de qualquer tipo de imagem. Nesta tese, propõe-se uma condição de contorno para que a qualidade possa ser realçada: deve ser possível reproduzir nos exemplos a mesma redução de qualidade da informação que se quer realçar, pois, para que um bom processo de casamento seja realizado, a redução de qualidade na informação exemplo deve ser reproduzido, de forma que o resultado tenha qualidade similar à informação de baixa qualidade a ser realçada. Além disso, a informação de realce tende a ser mais confiável quando a distância entre os casamentos é menor.

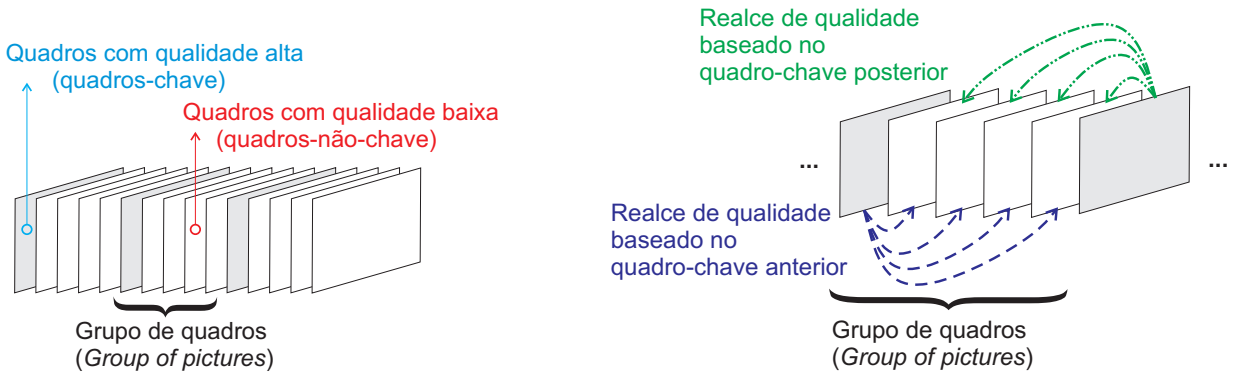
5.2 CODIFICAÇÃO E REALCE DE VÍDEO COM QUALIDADE DE COMPRESSÃO MISTA

Para codificar um vídeo com qualidade de codificação mista, deve-se variar temporalmente os parâmetros de quantização, vide Figura 5.2(a), e determinar um tamanho para o GOP, que é o menor conjunto de quadros que represente a codificação total da seqüência. Neste caso, teremos dois tipos de quadros ao variarmos o parâmetro de quantização (Q): um de melhor qualidade (Q_{chave}) e outro de qualidade reduzida ($Q_{não\text{-}chave}$), ou seja, $Q_{chave} < Q_{não\text{-}chave}$.

A decodificação é feita da maneira tradicional (com o decodificador padrão), e em seguida são adicionadas aos quadros-não-chave as informações de realce (na grande maioria de alta freqüência) que se encontram nos quadros-chave, como mostra a Figura 5.2(b). Observe que são usados os quadros de um GOP acrescido do quadro-chave posterior. Assim, o realce bidirecional é realizado ao utilizarmos dois quadros-chave (um anterior e um posterior) para cada quadro não-chave.

A Figura 5.2 mostra um esquemático simplificado da aplicação do processo de realce na arquitetura proposta. Esta tese apresenta um método para estimação de informações perdidas (quantizadas com $Q_{não\text{-}chave}$) dos quadros-não-chave que estão contidos nos quadros-chave (que mantém uma qualidade maior, pois foram quantizadas com $Q_{chave} < Q_{não\text{-}chave}$.

O método de realce proposto é baseado no trabalho de Brandi *et al*. [95], onde é feita uma super-resolução de uma seqüência de vídeo baseado em quadros-chave. Neste trabalho, ao invés de serem utilizados quadros-não-chave com resolução reduzida, os quadros-não-chave passam por uma quantização



(a) (b)

Figura 5.2: Vídeo contendo quadros de diferentes qualidades. (a) Codificação dos quadros-chave e não-chave com diferentes parâmetros de quantização. (b) Decodificador com o realce dos quadros-não-chave utilizando os quadros-chave.

maior (que a dos quadros-chave), o que implica em uma redução de qualidade. A princípio o processo de decodificação de um vídeo comprimido com qualidade mista pode ser feito com um decodificador padrão, e o processo de realce poderá ser adicionado ao processo de decodificação de forma a aumentar a qualidade dos quadros-não-chave (vide Figura 5.2(b)). A Figura 5.3 mostra um diagrama de blocos simplificado do processo de realce proposto.

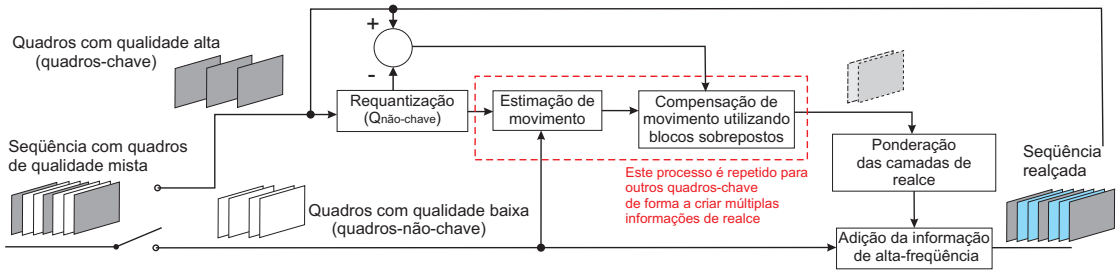


Figura 5.3: Diagrama de blocos do realce de vídeos com qualidade mista.

Primeiramente, utiliza-se a informação do GOP para distinguir entre os quadros-chave e não-chave. Os quadros-chave são transformados e re-quantizados (utilizando $Q_{não\text{-}chave}$) e em seguida reconstruídos, gerando assim uma versão de baixa qualidade do quadro-chave. A extração da informação para o melhoramento da qualidade é realizada a partir da diferença entre o quadro-chave e o quadro-chave com a qualidade reduzida. Em seguida fazemos uma estimação de movimento bidirecional entre as versões de baixa qualidade dos quadros-chave e o quadro não-chave. Já a compensação de blocos com sobreposição utiliza os vetores de movimento do processo de estimação de movimento, juntamente com a informação de realce extraída dos quadros-chave, o que permite a criação de uma camada de realce

contendo as informações de alta-freqüência espacialmente registradas e compatíveis com o conteúdo de baixa qualidade. Os quadros-não-chave realçados são obtidos ao adicionar a camada de realce no quadro de baixa qualidade.

5.3 VÍDEO COM FOCO MISTO E VÍDEO COM RUÍDO MISTO

Diferentes tipos de capturas desfocadas podem ocorrer em um vídeo, como por exemplo: o desfoque gaussiano, o desfoque de movimento, o desfoque radial e etc [112]. Estes resultados são decorrentes do uso de diferentes parâmetros da câmera, tais como a abertura do diafragma, a distância focal, tipo de lentes, tempo de exposição e etc, além da distância e movimento entre a câmera e os objetos em cena. Entretanto, nem todas estas distorções do processo de captura são desejáveis. Por exemplo, em câmeras de vídeo não profissionais, quadros com diferentes focos podem ocorrer devido ao atraso nos componentes mecânicos durante a atuação do autofoco. O resultado é uma seqüência de vídeo com foco misto, onde alguns quadros estão com o foco normal enquanto outros estão desfocados. Nesta tese o realce baseado em exemplos é aplicado nos quadros desfocados de forma a aumentar a sua qualidade objetiva e subjetiva.

Outro cenário onde a captura do vídeo é ruidosa, do tipo '*salt and pepper*', cujos quadros-não-chave possuem aleatoriamente pixels espúrios de tonalidades clara ou escura, também será abordada nesta tese. Este tipo de ruído pode ser causado por conversores analógico-digital ou erros em alguns bits de transmissão [113]. Note que, neste caso, o processo de realce proposto na Figura 5.1 dificilmente gerará bons resultados, já que o quadro-não-chave ruidoso não terá correlação com os dicionários compostos por blocos ruidosos e realces também ruidosos. Pode-se então concluir que para um bom funcionamento do processo de realce baseado em exemplos, deve-se utilizar um dicionário cujos blocos possam ser processados de forma a gerar blocos correlacionados com a informação que se deseje realçar.

Portanto, para contornar este problema, propõe-se a utilização do filtro da mediana para reduzir o ruído '*salt and pepper*', como descreve a literatura [56]. Assim, um vídeo com ruído '*salt and pepper*' misto é transformado em um vídeo com filtragem da mediana mista, o que viabiliza o uso do processo de realce baseado em exemplos.

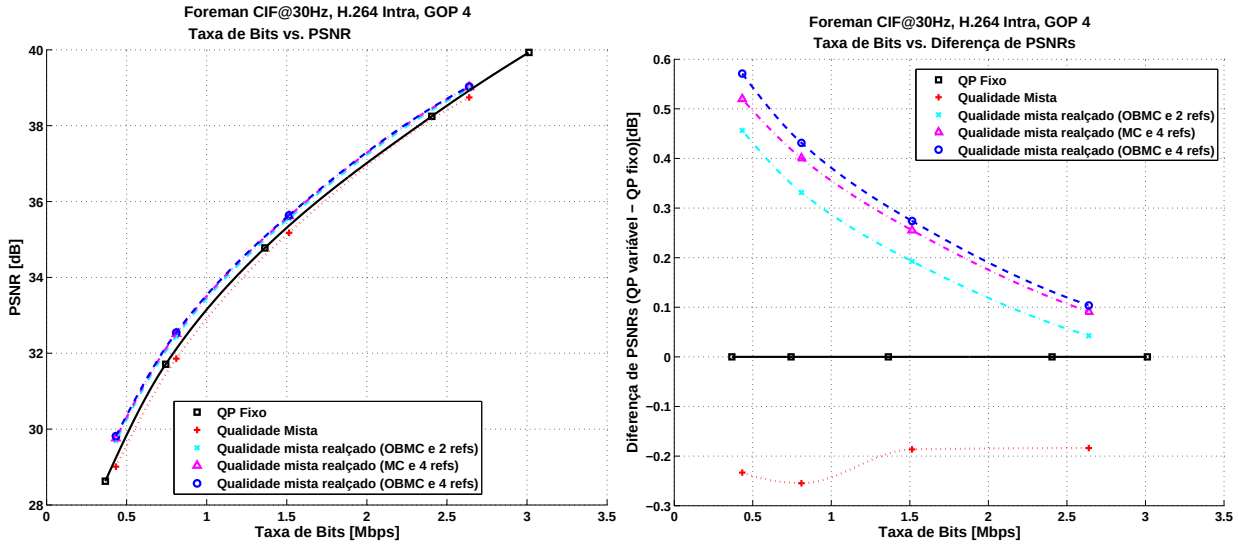
5.4 RESULTADOS EXPERIMENTAIS DO REALCE UTILIZANDO QUALIDADE DE COMPRESSÃO MISTA

O desempenho do realce em vídeos contendo quadros com qualidade mista foi realizado utilizando seqüências de vídeo de tamanho CIF e 720p, codificadas com H.264-Intra (JM 15.1) com GOP igual a 4 (ou seja, para cada quadro-chave, existem três quadros-não-chave). Os parâmetros de quantização utilizados foram $\{22, 27, 32, 37\}$, de forma a gerar uma curva que relaciona taxa e distorção. Definimos ainda que $Q_{\text{não-}chave} = 2Q_{chave}$, i.e. $QP_{\text{não-}chave} = QP_{chave} + 6$ [4, 5]. A estimação de movimento no processo de realce utiliza a busca completa por uma janela de 32×32 pixels para os macroblocos e sub-macroblocos.

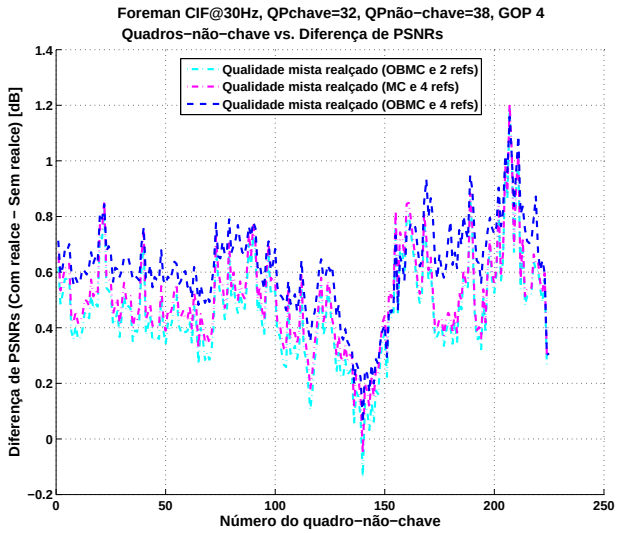
A Figura 5.4(a) mostra o desempenho do H.264 utilizando apenas quadros do tipo *intra* e QP fixo, que é comparado com diversas configurações de realce. Para as traçar as curvas, escolhemos os pontos da codificação com o QP fixo que mais se aproximam dos pontos resultantes da codificação com QPs mistos. Testes foram feitos utilizando dois quadros de referências (os quadros-chave anterior e posterior mais próximos) e com quatro quadros-chave (dois anteriores e dois posteriores). Comparamos também os resultados da compensação de movimento utilizando blocos de tamanho variáveis sobrepostos (OBMC) com a técnica de compensação de movimento ordinária (MC). A Figura 5.4(b) é a versão diferencial da Figura 5.4(a), onde a curva de taxa-distorção do caso onde o QP é fixo foi utilizado como referência. Apesar da perda de desempenho do vídeo com QPs mistos (comparado com QP fixo), podemos obter ganhos significativos de taxa-distorção ao aplicarmos o processo de realce proposto nesta tese.

Na Figura 5.4(c) mostramos os resultados quadro-a-quadro para a seqüência *Foreman* codificado com $Q_{chave} = 32$ e $Q_{\text{não-}chave} = 38$. Neste caso, com duas referências e OBMC, o ganho médio foi de 0,49 dB. Com quatro referências e MC a média aumenta para 0,87dB. Por fim, ganhos de 0,91 dB são obtidos ao utilizarmos quatro quadros de referência e OBMC. Apesar dos ganhos serem modestos, podemos mostrar na Figura 5.5 um ganho de qualidade significativo, onde colocamos o 51^o quadro-não-chave do vídeo *Foreman* sem realce (Fig. 5.5(a)) e com realce (Fig. 5.5(c)).

A Figura 5.6(a) mostra a comparação entre os métodos propostos e a compressão de vídeo com parâmentro de qualidade fixa aplicados às seqüências contendo pouco movimento. As Figs. 5.6(b) e (c) mostram os resultados diferenciais para uma seqüência de contendo pouco e muito movimento, respectivamente. Já a Figura 5.7(a) mostra as curvas de taxa-distorção para a seqüência Shields, as Figs 5.7(b) e (c) mostram as curvas diferenciais para as seqüências de vídeo de alta-resolução.



(a) (b)



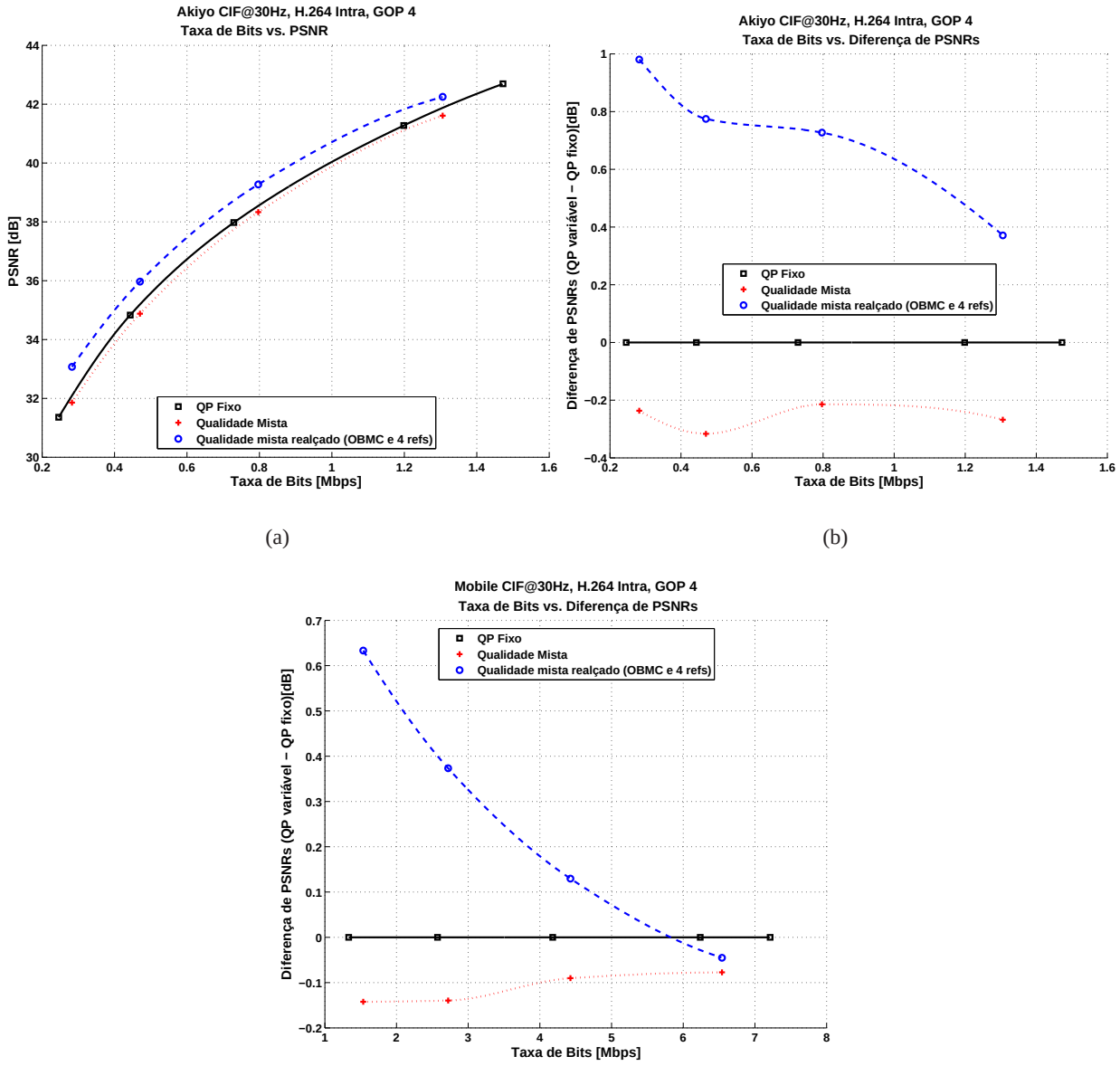
(c)

Figura 5.4: Resultados da codificação da seqüência *Foreman*, comparando os vídeos codificados utilizando H.264 *intra* com: QP fixo, QPs variáveis e vídeos com QPs variáveis realçados (utilizando OBMC com 2 e 4 referências ou dicionários, e utilizando MC com 4 referências). (a) Curvas de taxa-distorção. (b) Curva diferencial de (a), tendo como referência o vídeo com QP fixo. (c) Comparação quadro-a-quadro para a seqüência *Foreman* codificado com Q_{chave}=32, $Q_{não\text{-}chave}$=38.

(a)

(b)

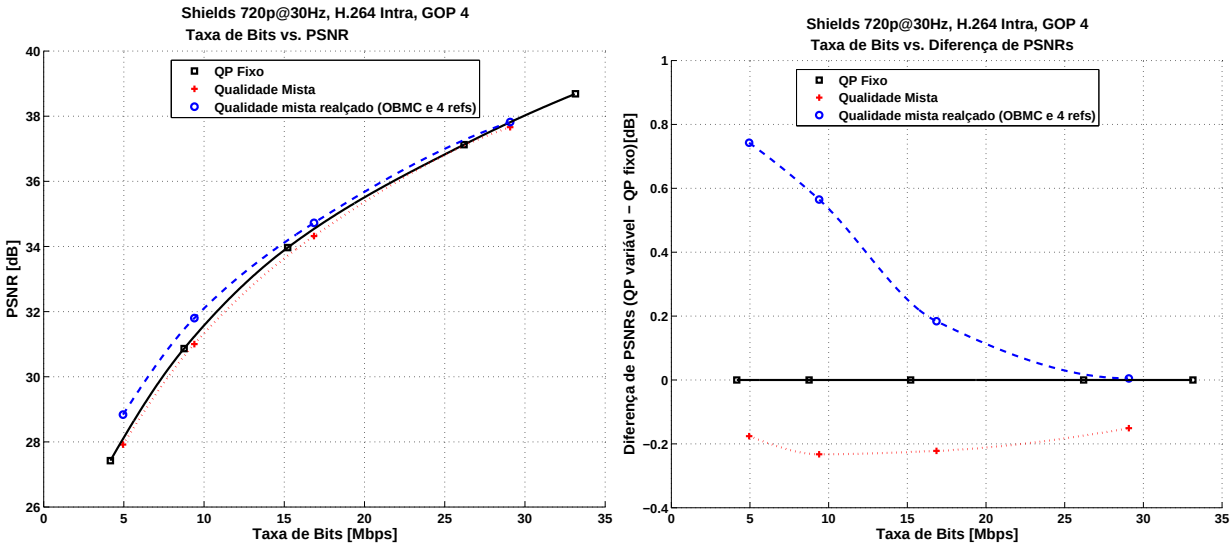
Figura 5.5: Comparação subjetiva do proposto realce de qualidade baseado em exemplos. (a) Quadro de baixa qualidade (não-chave) e (b) quadro não-chave realçado. Seqüência *Foreman* codificada com Q_{chave}=32, $Q_{não\text{-}chave}$=38 e GOP igual a 4.



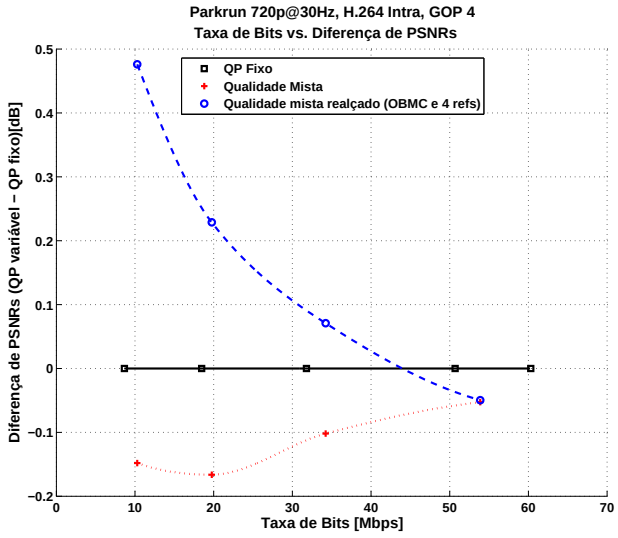
(a) (b)

(c)

Figura 5.6: Resultados comparando H.264 intra com parâmetro de qualidade fixo, com parâmetro de qualidade variável e com parâmetro de qualidade variável após o processo de realce utilizando a seqüência *Akiyo*. (a) curvas taxa-distorção. (b) O gráfico anterior com curvas diferenciais. (c) Curvas de taxa-distorção diferenciais para a seqüência de vídeo *Mobile* que compara o vídeo codificado com QP fixo, QP variável e QP variável com realce.



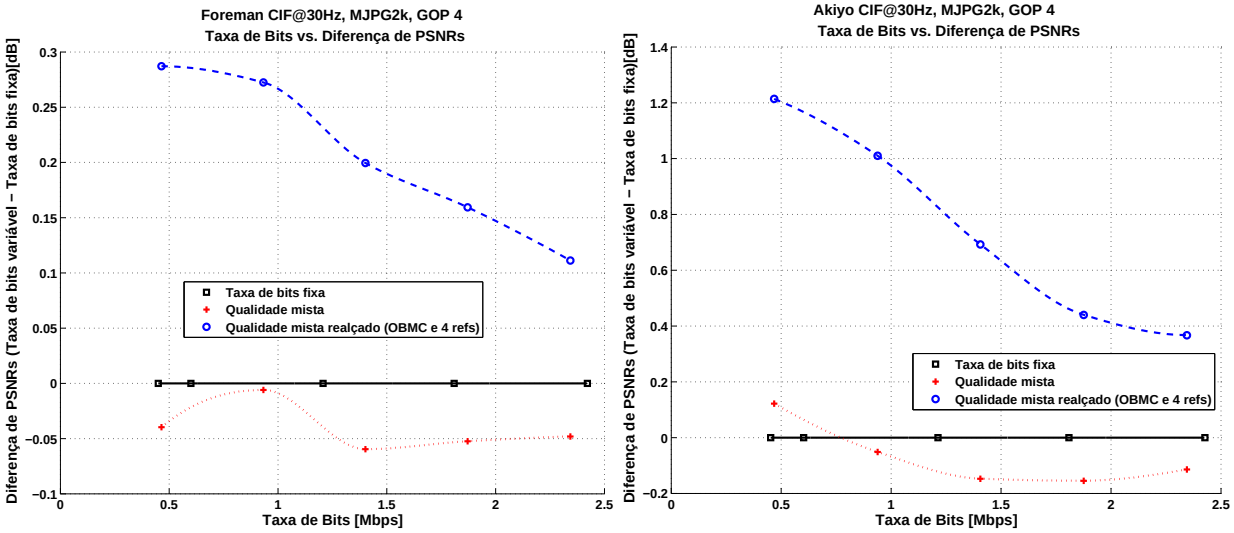
(a) (b)



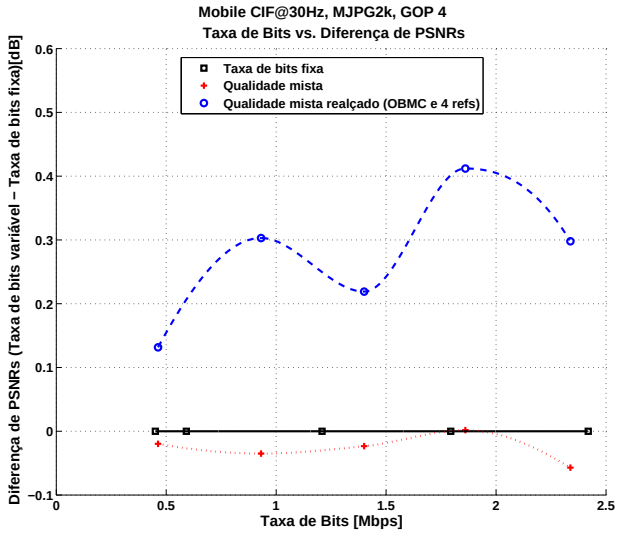
(c)

Figura 5.7: Resultados comparando H.264 intra com parâmetro de qualidade fixo, com parâmetro de qualidade variável e com parâmetro de qualidade variável após o processo de realce utilizando a seqüência *Shields*. (a) curvas taxa-distorção. (b)(c) Curvas de taxa-distorção diferenciais que compara o vídeo codificado com QP fixo, QP variável e QP variável com realce utilizando as seqüências de vídeo *Shields* e *Parkrun*, respectivamente.

Testes utilizando o Motion JPEG 2000 (implementados com o software Kakadu [114]) foram realizados em vídeos no formato CIF. Neste caso, ao invés de determinar um parâmetro de quantização fixo, iremos fixar a taxa de bits para cada quadro. No caso, a relação de taxa entre os quadros de qualidade baixa (quadros-não-chave) e qualidade alta (quadros-chave) foi definida como $7/10$. Como mostrado nas Figuras 5.8(a)-5.8(c), observe que é possível aumentar o desempenho de taxa-distorção após o processo de realce.



(a) (b)

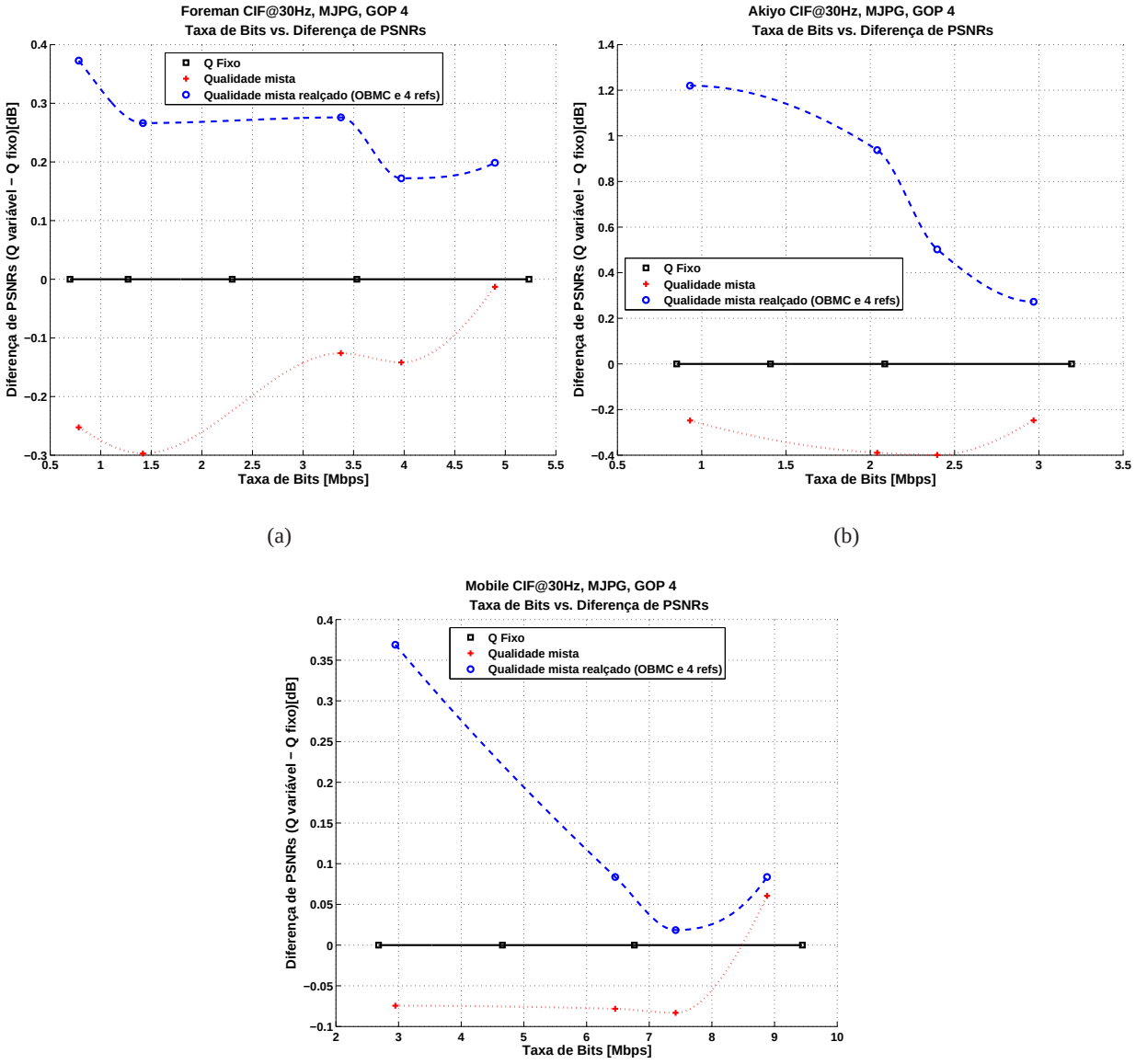


(c)

Figura 5.8: Curvas diferenciais comparando o desempenho do Motion JPEG 2000 para bits-por-quadro fixa, bits-por-quadro misto e bits-por-quadro misto com o método de realce proposto. Os testes foram feitos com as seqüências (a) *Foreman*, (b) *Akiyo* e (c) *Mobile*.

Em seguida aplicamos o método de qualidade mista com o Motion JPEG. Onde os testes foram realizados utilizando matrizes de quantização três vezes maiores (em intensidade) nos quadros-não-chave

comparado com os quadros-chave. As Figuras 5.9(a)-(c) também mostram um ganho de desempenho ao se utilizar o método de qualidade mista com realce.



(a) (b)

(c)

Figura 5.9: Curvas de taxa-distorção diferenciais que compara o vídeo codificado com Motion JPEG utilizando matriz de quantização (Q) fixo, Q variável e Q variável com realce utilizando as seqüências de vídeo *Foreman*, *Akiyo* e *Mobile*, respectivamente.

Na Tabela 5.1, fazemos o uso da métrica objetiva [102] para calcular a redução na taxa de bits do método que utiliza quadros de qualidade (ou taxa) mista, comparados com vídeos comprimidos com parâmetro de qualidade ou taxa fixos. Os resultados mostram que uma perda em termos de taxa-distorção ocorre quando o sistema de qualidade (ou taxa) mista é utilizado. Entretanto, ao aplicar o realce proposto nesta tese, obtemos uma redução da taxa de bits. Observe ainda que a melhor configuração de realce ocorre quando utilizamos a compensação de movimento com sobreposição e a aplicação da estimação e

compensação de movimento utilizando múltiplos dicionários (que neste experimento utilizamos dois ou quatro quadros-chave para o realce dos quadros-não-chave).

Tabela 5.1: Redução da taxa de bits [102]

Seqüência de vídeo comprimida	Redução de taxa sobre vídeos com parâmetros fixos
Foreman $\rightarrow H.264_{MQ\,no\,enh.}$	-4.28%
Foreman $\rightarrow H.264_{MQ\,OBMC\,(2\,refs)}$	5.29%
Foreman $\rightarrow H.264_{MQ\,MC\,(4\,refs)}$	6.62%
Foreman $\rightarrow H.264_{MQ\,OBMC\,(4\,refs)}$	7.19%
Akyio $\rightarrow H.264_{MQ\,no\,enh.}$	-4.22%
Akyio $\rightarrow H.264_{MQ\,MC\,(4\,refs)}$	12.05%
Akyio $\rightarrow H.264_{MQ\,OBMC\,(4\,refs)}$	12.70%
Mobile $\rightarrow H.264_{MQ\,no\,enh.}$	-1.47%
Mobile $\rightarrow H.264_{MQ\,OBMC\,(4\,refs)}$	3.47%
Shields $\rightarrow H.264_{MQ\,no\,enh.}$	-4.08%
Shields $\rightarrow H.264_{MQ\,OBMC\,(4\,refs)}$	7.73%
Parkrun $\rightarrow H.264_{MQ\,no\,enh.}$	-2.04%
Parkrun $\rightarrow H.264_{MQ\,OBMC\,(4\,refs)}$	2.81%
Foreman $\rightarrow MJPG2k_{MQ\,no\,enh.}$	-0.84%
Foreman $\rightarrow MJPG2k_{MQ\,OBMC\,(4\,refs)}$	5.01%
Akyio $\rightarrow MJPG2k_{MQ\,no\,enh.}$	-0.95%
Akyio $\rightarrow MJPG2k_{MQ\,OBMC\,(4\,refs)}$	13.28%
Mobile $\rightarrow MJPG2k_{MQ\,no\,enh.}$	-0.61%
Mobile $\rightarrow MJPG2k_{MQ\,OBMC\,(4\,refs)}$	7.63%
Foreman $\rightarrow MJPG_{MQ\,no\,enh.}$	-4.22%
Foreman $\rightarrow MJPG_{MQ\,OBMC\,(4\,refs)}$	5.43%
Akyio $\rightarrow MJPG_{MQ\,no\,enh.}$	-3.58%
Akyio $\rightarrow MJPG_{MQ\,OBMC\,(4\,refs)}$	25.64%
Mobile $\rightarrow MJPG_{MQ\,no\,enh.}$	-0.48%
Mobile $\rightarrow MJPG_{MQ\,OBMC\,(4\,refs)}$	2.16%

5.5 RESULTADOS EXPERIMENTAIS DE REALCES EM VÍDEOS COM FOCO MISTO

Para testar o processo de realce numa seqüência de vídeo com focos temporalmente mistos, foi gerado sinteticamente um vídeo que simula quadros desfocados utilizando um filtro Gaussiano espacial. Os testes foram realizados com 300 quadros da seqüência *Shields*, onde um quadro focado ocorre periodicamente a cada 30 quadros. Para um filtro Gaussiano de tamanho 8×8, a média de PSNRs dos quadros-não-chave é de 25,62 dB. Após o processo de realce, a qualidade objetiva aumenta para 30,24 dB. No caso de utilizarmos um filtro Gaussiano de tamanho 5×5, o vídeo com foco misto tem em média 28,23 dB nos quadros desfocados e 32,19 dB depois do realce. O resultado subjetivo do processo de realce pode ser observado ao comparar uma região do quadro desfocado da Figura 5.10(a) com a mesma região após a utilização do realce proposto mostrada na 5.10(b). Ao aplicar este mesmo teste para outras seqüências, no formato CIF, tem-se o resultado mostrado na Tabela 5.2. O ganho objetivo médio do processo de realce neste conjunto de testes foi de 8,81 dB.

Tabela 5.2: Média das PSNRs dos quadros-não-chave: desfocados com o filtro Gaussiano 8×8 e realçados.

Seqüência	Média das PSNRs dos quadros-não-chave do vídeo desfocado	Média das PSNRs dos quadros-não-chave do vídeo realçado
Foreman	25,03 dB	31,30 dB
News	21,81 dB	33,18 dB
Mobile	17,73 dB	24,61 dB
Hall	22,07 dB	33,18 dB
Container	21,29 dB	33,92 dB
Shields	25,62 dB	30,24 dB
Média	22,26 dB	31,07 dB

5.6 RESULTADOS EXPERIMENTAIS DE REALCES EM VÍDEOS COM RUÍDO MISTO

Para o vídeo com ruído misto, foram inseridos nos quadros-não-chave um ruído '*salt and pepper*' em 2% dos pixels. Os quadros sem ruído utilizados como exemplo ocorrem no vídeo periodicamente a cada 30 quadros. O teste foi realizado em 300 quadros da seqüência *Shields* e a média de PSNRs nos quadros-

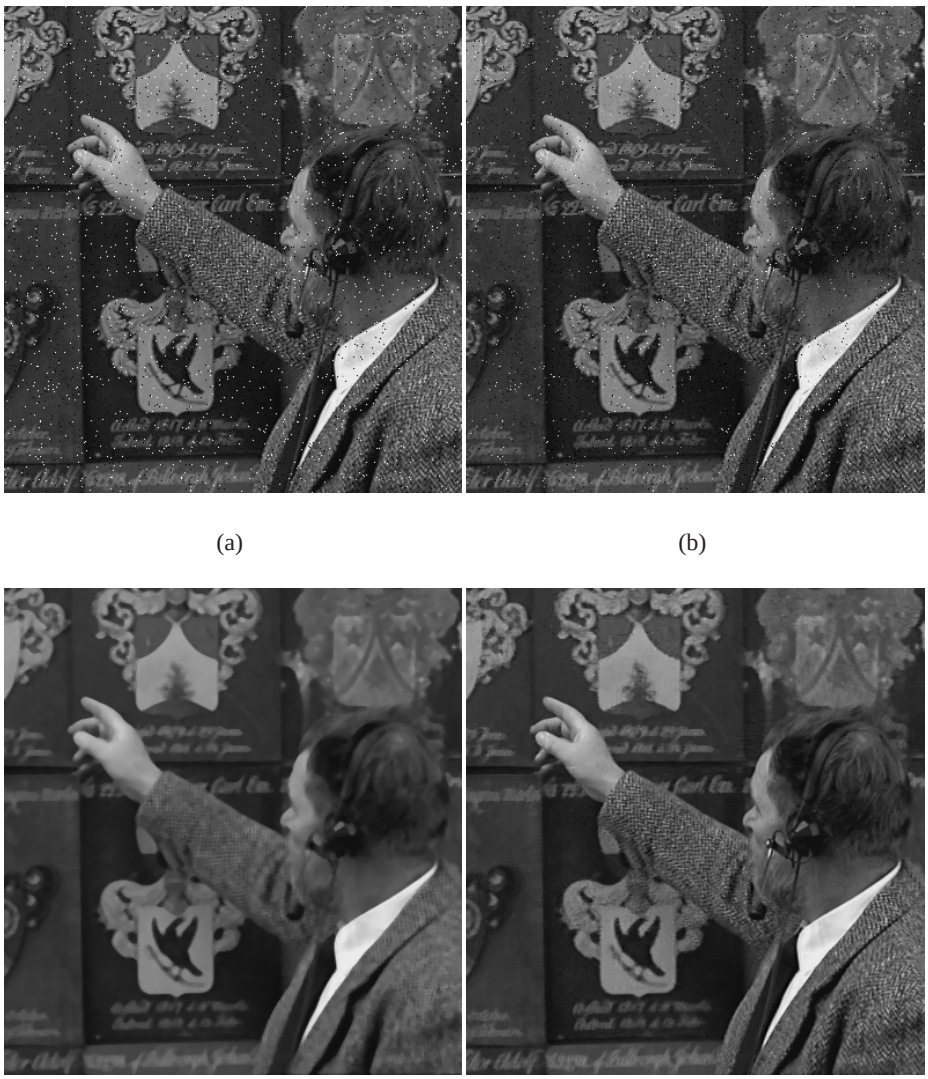
(a) (b)

Figura 5.10: Região do 16^o quadro da seqüência *Shields*: (a) filtrada com uma Gaussiana de tamanho 5×5 e (b) realçada baseada em exemplos.

não-chave é 21,78 dB. A Figura 5.11(a) exemplifica o ruído '*salt and pepper*' num quadro-não-chave. De acordo com o método de realce, uma degradação equivalente deve ser feita nos quadros-exemplo antes de realizar o casamento com o quadro-não-exemplo. Entretanto, devido à natureza aleatória do ruído, muitos descasamentos devem ocorrer no processo de estimação de movimento. Além disso, a informação de realce gerada é descorrelacionada com o quadro-não-exemplo. Consequentemente, o processo de realce pode aumentar o ruído ou gerar resultaddos pouco satisfatórios nos quadros-não-chave realçados, o que não ocorreu neste exemplo, já que o ruído utilizado é relativamente pequeno. O resultado do processo de realce aplicado ao ruído '*salt and pepper*' realçou em média os quadros-não-chave em 25,19 dB. Entretanto, como pode ser observado na Figura 5.11(b) o resultado do processo de realce não foi satisfatório.

Como descrito anteriormente, os filtros da mediana podem reduzir o ruído '*salt and pepper*'. Portanto, ao filtrar os quadros ruidosos com um filtro da mediana de tamanho 5×5 obtém-se uma PSNR média de 29,01 dB nos quadros-chave filtrados. Ao aplicar o processo de realce na seqüência filtrada, obtém-se uma PSNR média de 31,44 dB nos quadros realçados. Um exemplo da imagem filtrada pode ser observada na Figura 5.11(c) e o resultado do processo de realce na Figura 5.11(d). Observe que o uso do filtro da mediana no processo de realce do vídeo com ruído '*salt and pepper*' misto resulta em um desempenho objetivamente e subjetivamente melhor, se comparado com o realce sem o filtro da mediana.

A Tabela 5.3 mostra o desempenho do teste citado anteriormente aplicado em outras seqüências. Nele pode-se observar que o ganho médio dos quadros-não-chave após o processo de realce em vídeos com



(a) (b)

(c) (d)

Figura 5.11: Região do 16^o quadro da seqüência *Shields*: (a) com ruído '*salt and pepper*', (b) realçado utilizando exemplos com ruído '*salt and pepper*', (c) com um filtro da mediana aplicado ao ruído '*salt and pepper*', (d) realçada utilizando o filtro da mediana.

ruído '*salt and pepper*' é de 1,61 dB. Apesar de significativo, este ganho objetivo não resulta em resultados subjetivos satisfatórios, como mostra a Figura 5.11. Ao filtrar o quadro ruidoso com o filtro da mediana de 5×5, temos um ganho médio de 3,48 dB em comparação com o vídeo ruidoso original. Entretanto, seus resultados subjetivos geram imagens com pouco detalhamento, como exemplifica a Figura 5.11(c). Nesta tese, propõe-se realçar o vídeo filtrado, ao invés do vídeo ruidoso, permitindo assim um ganho objetivo de 9,93 dB se comparado com o vídeo com ruído '*salt and pepper*' misto.

Tabela 5.3: Média das PSNRs dos quadros-não-chave: com ruído '*salt and pepper*' (SP), com realce sobre o ruído '*salt and pepper*' (RSP), com ruído '*salt and pepper*' seguido da filtragem da mediana 5×5 e com realce sobre a filtragem da mediana 5×5.

Seqüência	Média das PSNRs dos quadros-não-chave do vídeo com ruído '*salt and pepper*'	Média das PSNRs dos quadros-não-chave do vídeo com realce sobre o ruído '*salt and pepper*'	Média das PSNRs dos quadros-não-chave do vídeo com ruído '*salt and pepper*' seguido da filtragem da mediana 5×5	Média das PSNRs dos quadros-não-chave do vídeo com realce sobre o filtro da mediana 5×5
Foreman	22,14 dB	24,97 dB	29,48 dB	33,35 dB
News	21,86 dB	25,11 dB	26,01 dB	35,25 dB
Mobile	22,05 dB	23,05 dB	19,43 dB	24,18 dB
Hall	22,37 dB	24,94 dB	25,80 dB	34,64 dB
Container	22,42 dB	24,83 dB	23,76 dB	33,31 dB
Shields	21,78 dB	25,19 dB	29,01 dB	31,44 dB
Média	22,10 dB	23.71 dB	25,58 dB	32,03 dB

6 CONCLUSÕES

Nesta tese, propomos alguns cenários de aplicações que permitem o uso de imagens correlacionadas e dicionários dinamicamente populados por meio da utilização da estimação de movimento para a busca de casamentos entre a informação a ser realçada com os exemplos. Foi proposto um método que permite usar, descartar ou fundir as informações de alta-freqüência de um conjunto de dicionários oriundos de outros quadros de referência, obtendo ganhos objetivos e subjetivos significativos. Outro método apresentado, realiza a sobreposição dos blocos durante a compensação de movimento, o que contribui para o aumento dos ganhos objetivos e a redução dos efeitos de blocos durante o processo de realce. Nos cenários de vídeo codificado com quadros de resolução mista, por exemplo, os vídeos de baixa-resolução com fotografias redundantes resultam em ganhos que podem chegar até a 3 dB após o processo de realce, quando comparado aos vídeos interpolados. No tratamento de vídeo com resolução mista, podemos diminuir a complexidade do codificador ao diminuir o esforço computacional do processo de estimação de movimento (que é feito em um quadro com resolução menor). Por outro lado, o cenário onde fotografias são tiradas enquanto um vídeo de baixa-resolução está sendo gravado permite uma extrapolação da resolução máxima do vídeo. Neste caso, o aumento de resolução do vídeo é baseado nas fotografias de resolução mais alta, utilizando a técnica de realce baseado em exemplos proposto nesta tese.

Um novo método de SR no domínio da transformada é apresentado, para uso em sistemas de resolução mista para seqüências com uma vista e também com múltiplas vistas (com informação de profundidade). Uma técnica baseada em DCT é introduzida para realizar a interpolação da imagem em baixa resolução. O método proposto procede projetando a vista em alta resolução para o ponto de vista da imagem de baixa resolução. Coeficientes de alta freqüência da vista projetada são utilizados para preencher os coeficientes de alta freqüência que estão ausentes na imagem em baixa resolução. A imagem submetida ao processo de SR no domínio da transformada alcança ganhos de qualidade significativos sobre a imagem interpolada, tanto em termos objetivos como subjetivos.

Em seguida, generalizamos o processo de realçar seqüências de vídeo que utilizam quadros com qualidade variável, utilizando uma arquitetura similar ao da super-resolução baseada em exemplos. Os experimentos mostram ainda que o método em questão funciona para vários tipos de *codecs* de vídeo, como por exemplo o H.264/AVC, Motion JPEG e Motion JPEG 2000, podendo reduzir o tamanho do vídeo codificado em qualidade mista em torno de 7% se comparamos com seqüências codificadas com

parâmetro de quantização fixo. Além disso, o realce proposto foi realizado em vídeos filtrados, ou fora de foco, e vídeo com ruído, obtendo bons resultados objetivos e subjetivos.

Trabalhos futuros incluem investigar a remoção no domínio da transformada de ruído, assim como a redução de artefatos e o aguçamento da imagem submetida à super-resolução por meio da manipulação dos componentes DCT de alta freqüência. A arquitetura proposta também permite o uso de outras técnicas de decomposição em freqüência além da DCT como, por exemplo, as *wavelets*. Pode-se também explorar diferentes mecanismos para a utilização dos quadros-chave, como em [115], onde o processo de estimação e extração da informação de alta-freqüência são feitos utilizando uma técnica de detecção de características relevantes (*features*) seguida da transformação invariante à escala, deslocamento e orientação. Desta maneira, substitui-se o uso da estimação e compensação de movimento baseada em blocos no processo de realce apresentado nesta tese. Sugere-se ainda o estudo sobre métricas sem referência de qualidade de imagens e vídeos para que se possa controlar a quantidade de informação de alta-freqüência que deve ser atribuído a um quadro ou bloco não-chave durante o processo de realce.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] YANG, J. et al. Image super-resolution as sparse representation of raw image patches. *IEEE Computer Vision and Pattern Recognition (CVPR)*, San Jose, EUA, p. 1–8, Junho 2008.

[2] ZEYDE, R.; ELAD, M.; PROTTER, M. On single image scale-up using sparse-representations. *Curves & Surfaces, Avignon-France*, p. 24–30, Junho 2010.

[3] RICHARDSON, I. E. G. *H.264 and MPEG-4 Video Compression*. [S.l.]: John Wiley & Sons Ltd, 2003.

[4] RICHARDSON, I. E. G. *The H.264 Advanced Video Compression Standard*. [S.l.]: John Wiley & Sons Ltd, 2010.

[5] SAYOOD, K. *Introduction to Data Compression*. [S.l.]: Morgan Kuffmann Publishers, 2005.

[6] MüLLER, K.; MERKLE, P.; WIEGAND, T. 3/-D Video Representation Using Depth Maps. *Proceeding of IEEE*, v. 99, n. 4, p. 643 – 656, 2011.

[7] HUNG, E. M. et al. Transform-domain super resolution for multiview images using depth information. *European Signal Processing Conference*, Barcelona, Spain, Agosto 2011.

[8] KAUFF, P. et al. Depth map creation and image-based rendering for advanced 3dtv services providing interoperability and scalability. *Image Communication*, 2007.

[9] VETRO, A. et al. 3d-tv content storage and transmission. *IEEE Transactions on Broadcasting*, v. 57, n. 2, p. 384–394, Junho 2011.

[10] CHEUNG, G.; VELISAVLJEVIC, V.; ORTEGA, A. On dependent bit allocation for multiview image coding with depth-image-based rendering. *IEEE Transactions on Image Processing*, v. 20, n. 11, p. 3179–3194, Novembro 2011.

[11] MACCHIAVELO, B.; MUKHERJEE, D.; QUEIROZ, R. L. D. Iterative side-information generation in a mixed resolution wyner-ziv framework. *IEEE Trans. Circuits and Systems for Video Technology*, v. 19, n. 10, p. 1409–1423, Outubro 2009.

[12] PEIXOTO, E.; QUEIROZ, R. L. de; MUKHERJEE, D. A wyner-ziv video transcoder. *IEEE Trans. Circuits and Systems for Video Technology*, v. 20, n. 2, p. 189–200, Fevereiro 2010.

[13] GARCIA, D. C.; DOREA, C. C.; QUEIROZ, R. L. de. Super-resolution for multiview images using depth information. *Proc. IEEE Intl. Conf. on Image Processing*, Hong Kong, China, v. 10, n. 2, p. 188–193, Setembro 2010.

[14] HUNG, E. M.; QUEIROZ, R. L. de. Blocking-effect reduction in a reversed-complexity video codec based on a mixed-quality framework. *Proc. of Intl. Telecommunication Symposium*, Manaus, Brazil, Setembro 2010.

[15] MACCHIAVELO, B. *Codificador Distribuído de Vídeo com complexidade variável a partir de codificação em resolução espacial mista*. Tese (Doutorado) — Universidade de Brasília, Grupo de Processamento Digital de Sinais, Departamento de Engenharia Elétrica., 2009.

[16] MUKHERJEE, D. *A robust reversed complexity Wyner-Ziv video codec introducing sign-modulated codes*. Palo Alto, CA, EUA, Maio 2006.

[17] MUKHERJEE, D.; MACCHIAVELLO, B.; QUEIROZ, R. L. de. A simple reversed complexity wyner-ziv video coding mode based on a spatial reduction framework. *Proc. SPIE Visual Communications and Image Processing*, Janeiro 2007.

[18] KWON, D. K.; SHEN, M. Y.; KUO, C. C. J. Rate control for h.264 video with enhanced rate and distortion models. *IEEE Trans. on Circuits and Systems for Video Technology*, v. 13, n. 5, p. 517 – 529, Maio 2007.

[19] KIM, S.; HO, Y. S. Rate control algorithm for h.264/avc video coding standard based on rate-quantization model. *International Conference on Multimedia and Expo*, v. 1, p. 165 – 168, Junho 2004.

[20] MA, S. et al. Rate control for advance video coding (avc) standard. *International Symposium on Circuits and Systems*, v. 2, p. 892–895, Maio 2003.

[21] HUNG, E. M. et al. Video super-resolution using codebooks derived from key frames. *submitted to IEEE Trans. on Circuits and Systems for Video Technology*, 2011.

[22] YEO, C.; TAN, W. T.; MUKHERJEE, D. Receiver error concealment using acknowledge preview (recap) - an approach to resilient video streaming. *Proc. Int. Conference on Acoustics, Speech and Signal Processing*, Taiwan, Abril 2009.

[23] SHARMA, G. *Digital Color Imaging Handbook*. [S.l.]: CRC Press, 2002.

[24] ITU-T (ITU-T T.81) and ISO/IEC JTC1. *Digital Compression and Coding of Continuous-Tone Sitll Images*. [S.l.], Setembro 1992.

[25] IMAGE Coding System: Core Coding System (JPEG2000 Part 1). [S.l.], Setembro 2000.

[26] ITU-T. *Video Codec for Audiovisual Services at px64 kbit/s*. [S.l.], Version 1, Novembro 1990; Version 2, Março 1993.

[27] ITU-T and ISO/IEC JTC 1 - ISO/EC 13818-2 (MPEG-2). *Generic coding of moving pictures and associating audio information - Part 2: Video*. [S.l.], Novembro 1994.

[28] ITU-T. *ITU-T Recommendation H.263, Video coding for low bit rate communication*. [S.l.], Novembro 2000.

[29] ITU-T. *Joint Model Number 1 (JM-1) - JVT-A003*. [S.l.], Janeiro 2005.

[30] JVT of ISO/IEC MPEG and ITU-T VCEG. *Advanced Video Coding for Generic Audiovisual Services*. [S.l.], Março 2005.

[31] ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4). *Advanced Video Coding for Generic Audiovisual Services*. [S.l.], Março 2010.

[32] SHANNON, C. E. A mathematical theory of communication. *Bell Syst. Tech.*, n. J.27, p. 379–423, 623–656, 1948.

[33] HUFFMAN, D. A method for the construction of minimum redudancy codes. *Proc. IRE*, n. 40, p. 1098–1101, 1952.

[34] RAO, K. R.; HWANG, J. J. *Techniques and Standards for Image, Video and Audio Coding*. [S.l.]: Prentice Hall, 1997.

[35] MARPE, D. et al. Video compression using context-based adaptive arithmetic coding. *International Conference on Image Processing*, Thessaloniki (GR), p. 558–561, Setembro 2001.

[36] JBIG2. *JBIG2*. [S.l.], Abril 2001.

[37] ZAGHETTO, A. *Compressão de Documentos Compostos usando H.264/AVC-Intra*. Tese (Doutorado) — Universidade de Brasília, Grupo de Processamento Digital de Sinais, Departamento de Engenharia Elétrica., 2009.

[38] IMAGE Coding System Motion JPEG 2000 (JPEG2000 Part 3). [S.l.], Setembro 2003.

[39] KOGA, T. et al. Motion-compensated interframe coding for video conferencing. *Proc. NTC*, p. 961–965, Novembro 1981.

[40] LI, R.; ZENG, B.; LIOU, M. L. A new three-step search algorithm for block estimation. *IEEE Transactions on Circuits and Systems for Video Technologies*, v. 4, p. 438–443, Agosto 1994.

[41] PO, L. M.; MA, W. C. A novel four-step search algorithm for fast block estimation. *IEEE Transactions on Circuits and Systems for Video Technologies*, v. 6, p. 313–317, Junho 1996.

[42] THAM, J. Y.; RANGANATH, S.; KASSIM, A. A. A novel unrestricted center-biased diamond search algorithm for block motion estimation. *IEEE Transactions on Circuits and Systems for Video Technologies*, v. 8, n. 4, p. 369–377, Agosto 1998.

[43] ZHU, S.; MA, K. A new diamond search algorithm for fast block matching motion estimation. *ICICS'97*, Sinagpore, p. 9–12, Setembro 1997.

[44] TOURAPIS, A.; AU, O. C.; LIOU, M. L. Highly efficient predictive zonal algorithm for fast block-matching motion estimation. *IEEE Trans. Circuits and Systems for Video Technology*, v. 12, p. 934–947, Outubro 2002.

[45] HUNG, E. M.; QUEIROZ, R. L. de; MUKHERJEE, D. On macroblock partition for motion compensation. *International Conference on Image Processing*, Atlanta, EUA, p. 1697–1700, Outubro 2006.

[46] HUNG, E. M. *Compensação de Movimento Utilizando Blocos Multi-Escala e Forma Variável Em Um Codec de Vídeo Híbrido*. Tese (Doutorado) — Universidade de Brasília, Grupo de Processamento Digital de Sinais, Departamento de Engenharia Elétrica., 2007.

[47] FERREIRA, R. U. et al. Efficiency improvements for a geometric-partition-based video coder. *International Conference on Image Processing*, Cairo, Egito, Novembro 2009.

[48] SULLIVAN, G.; WIEGAND, T.; OHM, J. R. [S.l.]. Http://www.itu.int/en/ITU-T/studygroups/com16/video/Pages/jctvc.aspx.

[49] LUTHRA, A.; SULLIVAN, G. J.; WIEGAND, T. H.264/AVC Video Coding Standard. *IEEE Transactions on Circuits and Systems for Video Technologies*, v. 13, n. 7, Julho 2003.

[50] ARCHIBALD, R.; GELB, A. A method to reduce the gibbs ringing artifact in mri scans while keeping tissue boundary integrity. *IEEE Transactions on Medical Imaging*, v. 21, p. 305–319, Abril 2002.

[51] MALVAR, H. S. et al. Low-complexity transform and quantization in h.264/avc. *IEEE Trans. Circuits and Systems for Video Technology*, v. 13, p. 598–603, Julho 2003.

[52] MARPE, D.; SCHWARZ, H.; WIEGAND, T. Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard,. *IEEE Transactions on Circuits and Systems for Video Technology*, v. 13, n. 7, p. 620–636, Julho 2003.

[53] LIST, P. et al. Adaptive deblocking filter. *IEEE Transactions on Circuits and Systems for Video Technology*, v. 13, n. 7, p. 614–619, Julho 2003.

[54] WANG, Z.; LU, L.; BOVIK, A. C. Video quality assessment based on structural distortion measurement. *Signal Processing: Image Communication, special issue on "Objective video quality metrics"*, v. 19, n. 2, p. 121–132, Fevereiro 2004.

[55] WANG, Z.; BOVIK, A. C. Mean squared error love it or leave it? - a new look at signal fidelity measures. *IEEE Signal Processing Magazine*, v. 26, n. 1, p. 98–117, Janeiro 2009.

[56] PRATT, W. K. *Digital Image Processing: PIKS Inside*. California, EUA: Wiley-Interscience, 2001.

[57] GETREUER, P. Linear methods for image interpolation. *Image Processing On Line*.

[58] KEYS, R. Cubic convolution interpolation for digital image processing. *IEEE Trans. Acoustics, Speech, and Signal Processing*, v. 29, n. 6, p. 1153–1160, 1981.

[59] DUCHON, C. E. Lanczos filtering in one and two dimensions. *Journal of Applied Meteorology*, v. 18, n. 8, p. 1016–1022, Agosto 1979.

[60] PARK, S.; PARK, M.; KANG, M. Super-resolution image reconstruction a technical overview. *IEEE Signal Processing Magazine*, v. 20, n. 3, p. 21–36, Maio 2003.

[61] CHAUDHURI, S. *Super-Resolution Imaging*. [S.l.]: Kluwer, 2001.

[62] BARRETT, H. H.; MYERS, K. J. *Foundations on Image Science*. [S.l.]: John Wiley & Sons Ltd, 2004.

[63] SCHULTZ, R. R.; MENG, L.; STEVENSON, R. L. Subpixel motion estimation for super-resolution image sequence enhancement. *Journal of Visual Communication and Image Representation*, v. 9, n. 1, p. 38–50, Março 1998.

[64] BORMAN, S.; STEVENSON, R. L. Simultaneous multi-frame map super-resolution video enhancement using spatio-temporal priors. *IEEE International Conference on Image Processing*, v. 3, p. 469–473, Agosto 1998.

[65] KATSAGGELOS, A. K. Iterative image restoration algorithms. *Optical Engineering, special issue on Visual Communications and Image Processing*, v. 28, n. 7, p. 735–748, julho 1989.

[66] BOVIK, A. *Handbook of Image and Video Processing*. [S.l.]: Academic Press, 2000.

[67] SRIVASTAVA, A. Stochastic models for capturing image variability. *IEEE Signal Processing Magazine*, v. 19, n. 5, p. 63–76, Setembro 2002.

[68] WINKLER, G. *Image analysis, random fields and Markov chain Monte Carlo methods: a mathematical introduction*. [S.l.]: Springer-Verlag, 2003.

[69] STEVENSON, R. L.; SCHMITZ, B. E.; DELP, E. J. Discontinuity preserving regularization of inverse visual problems. *IEEE Transactions on System, Man and Cybernetics*, v. 24, n. 3, p. 455–469, Março 1994.

[70] BOUMAN, C.; SAUER, K. A generalized gaussian image model for edge-preserving map estimation. *IEEE Transactions on Image Processing*, v. 2, n. 3, p. 269–310, Julho 1993.

[71] HAMZA, A. B.; KOMATSU, T.; SAITO, T. Unifying probabilistic and variational estimation. *IEEE Signal Processing Magazine*, v. 19, n. 5, p. 37–47, Setembro 2002.

[72] ZIBETTI, M. V. W. *Super-resolução simultânea para seqüência de imagens*. Tese (Doutorado) — Universidade Federal de Santa Catarina, 2007.

[73] WANG, Z.; QI, F. Simultaneous multi-frame map super-resolution video enhancement using spatio-temporal priors. *IEEE Signal Processing Letters*, v. 11, n. 8, p. 678–681, Agosto 2004.

[74] KYBIC, J.; BLU, T.; UNSER, M. Generalized sampling a variational approach .i. theory. *IEEE Transactions on Signal Processing*, v. 50, n. 8, p. 1965–1976, Agosto 2002.

[75] KYBIC, J.; BLU, T.; UNSER, M. Generalized sampling a variational approach .ii. applications. *IEEE Transactions on Signal Processing*, v. 50, n. 8, p. 1977–1985, Agosto 2002.

[76] BENEDETTO, J.; ZAYED, A. *Sampling, Wavelets, and Tomography (Applied and Numerical Harmonic Analysis Series)*. [S.l.]: Birkhäuser, 2004.

[77] SHIN, J.; CHOUNG, Y. C.; PAIK, J. High-resolution image sequence interpolation. *IEEE Region 10 Annual Conference on Speech and image technologies for computing and telecommunications*, v. 2, p. 781–784, 1997.

[78] KOO, Y.; KIN, W. An image resolution enhancing technique using adaptive sub-pixel interpolation for digital still camera system. *IEEE Transactions on Consumer Electronics*, v. 45, n. 1, p. 118–123, Fevereiro 1999.

[79] IRANI, M.; PELEG, S. Motion analysis for image enhancement Resolution, occlusion, and transparency. *Journal of Visual Communication and Image Representation*, v. 4, n. 4, p. 324–335, Dezembro 1993.

[80] ZOMET, A.; RAV-ACHA, A.; PELEG, S. Robust super-resolution. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, v. 1, n. 1, p. 645–650, 2001.

[81] JIANG, Z.; WONG, T. T.; BAO, H. Practical super-resolution from dynamic video sequences. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, v. 2, n. 1, p. 549–554, 2003.

[82] SAAD, Y. *Iterative methods for sparse linear systems*. New Jersey, EUA: International Thompson Publishing, 1995.

[83] KATSAGGELOS, A. K.; MOLINA, R.; MATEOS, J. *Super Resolution of Images and Video* - Synthesis Lectures on Image, Video and Multimedia Processing. [S.l.]: Morgan and Claypool Publishers, 2007.

[84] TEKALP, A. M. *Digital Video Processing*. New Jersey, EUA: Prentice-Hall, 1995.

[85] TEKALP, A. M.; OZKAN, M.; SEZAN, M. I. High-resolution image reconstruction from lower-resolution image sequences and space-varying image restoration. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, v. 3, p. 169–172, 1992.

[86] EREN, P. E.; SEZAN, M. I.; TEKALP, A. M. Robust, object-based high-resolution image reconstruction from low-resolution video. *IEEE Transactions on Image Processing*, v. 6, n. 10, p. 1446–1451, Outubro 1997.

[87] PATTI, A. J.; SEZAN, M. I.; TEKALP, A. M. Robust methods for high-quality stills from interlaced video in the presence of dominant motion. *IEEE Transactions on Circuits and Systems for Video Technology*, v. 7, n. 2, p. 328–342, Abril 1997.

[88] WHEELER, F. W.; HOCTOR, R. T.; BARRETT, E. B. Super-resolution image synthesis using projections onto convex sets in the frequency domain. *IS&T/SPIE Symposium on Electronic Imaging, Conference on Computational Imaging*, San Jose, EUA, v. 5674, p. 479–490, Janeiro 2005.

[89] SEGALL, C. A. et al. Bayesian resolution enhancement of compressed video. *IEEE Transactions on Image Processing*, v. 13, n. 7, p. 21–36, 2004.

[90] TIPPING, M. E.; BISHOP, C. M. Bayesian image super-resolution. *In Advances in Neural Information Processing Systems*, v. 15, p. 1303–1310, 2002.

[91] FREEMAN, W. T.; PASZTOR, E. C.; CARMICHAEL, O. T. Learning low-level vision. *International Journal of Computer Vision*, p. 1–8, 2000.

[92] FREEMAN, W. T.; JONES, T. R.; PASZTOR, E. C. Example-based super-resolution. *IEEE Computer Graphics and Applications*, v. 22, p. 56–65, 2002.

[93] PROTTER, M.; ELAD, M. Image sequence denoising via sparse and redundant representations. *IEEE Transactions on Image Processing*, v. 18, n. 1, p. 27–36, Janeiro 2009.

[94] YANG, J. et al. Image super-resolution via sparse representation. *IEEE Transactions on Image Processing*, v. 19, n. 11, p. 2861–2873, Novembro 2010.

[95] BRANDI, F.; QUEIROZ, R. de; MUKHERJEE, D. Super-resolution of video using key-frames and motion estimation. *Proc. IEEE Intl. Conf. on Image Processing*, San Diego, CA, EUA, Outubro 2008.

[96] BRANDI, F.; QUEIROZ, R. de; MUKHERJEE, D. Super resolution of video using key frames. *Proc. IEEE Intl. Symp. on Circuits and Systems*, Seattle, EUA, Maio 2008.

[97] OLIVEIRA, K. F. et al. Bipredictive video super–resolution using key–frames. *Proc. IS&T/SPIE Symp. on Electronic Imaging, Visual Information Processing and Communication*, San Jose, CA, EUA, Janeiro 2010.

[98] SULLIVAN, G. J. et al. Overview of the H.264/AVC Video Coding Standard. *IEEE Transactions on Circuits and Systems for Video Technologies*, v. 13, Julho 2003.

[99] NOGAKI, S.; OHTA, M. An overlapped block motion compensation for high quality motion picture coding. *Proc. IEEE Int. Symp. Circuits Systems*, v. 1, p. 184–187, Setembro 1992.

[100] ZHANG, J.; AHMAD, M. O.; SWAMY, M. N. S. Overlapped variable size block motion compensation. *Proc. IEEE Intl. Conf. on Image Processing*, Santa Barbara, CA, EUA, v. 1, p. 184–187, Outubro 1997.

[101] SONG, B. C.; JEONG, S. C.; CHOI, Y. Video super-resolution algorithm using bi-directional overlapped block motion compensation and on-the-fly dictionary training. *IEEE Trans. Circuits and Systems for Video Technology*, v. 12, n. 3, Março 2011.

[102] BJONTEGAARD, G. *Calculation of Average PSNR Differences between RD curves*. Austin, Texas, EUA, Abril 2001.

[103] TRANSFORM-DOMAIN semi-super resolution. *International Conference on Image Processing*, Brussels, Belgium, Setembro 2011.

[104] DUGAD, R.; AHUJA, N. A fast scheme for image size change in the compressed domain. *IEEE Trans. Circuits and Systems for Video Tech.*, v. 11, n. 4, p. 461–474, Abril 2001.

[105] WU, Z.; YU, H.; CHEN, C. W. A new hybrid dct-wiener-based interpolation scheme for video intra frame up-sampling. *IEEE Signal Processing Letters*, v. 17, n. 10, p. 827–830, Outubro 2010.

[106] PERKINS, M. G. Data compression of stereopairs. *IEEE Trans. on Communications*, Potsdam, Alemanha, v. 40, p. 686–696, Maio 1992.

[107] BRUST, H. et al. Mixed resolution coding of stereoscopic video for mobile devices. *Proc. 3DTV Conference*, Potsdam, Alemanha, Maio 2009.

[108] JULESZ, B. *Foundations of cyclopean perception*. [S.l.]: University of Chicago Press, 1971.

[109] STELMACH, L. et al. Stereo image quality: effects of mixed spatio-tamporal resolution. *IEEE Trans. Circuits and Systems for Video Tech.*, v. 10, n. 2, p. 188–193, Março 2000.

[110] ZITNICK, C. et al. High-quality video view interpolation using a layered representation. *ACM SIGGRAPH*, Los Angeles, EUA, Agosto 2004.

[111] GARCIA, D. C. *Técnicas de Super-Resolução para Sistemas de Vídeo de Múltiplas Vistas em Resolução Mista*. Tese (Doutorado) — Universidade de Brasília, Grupo de Processamento Digital de Sinais, Departamento de Engenharia Elétrica., 2012.

[112] ENS, J.; LAWRENCE, P. An investigation of methods for determing depth from focus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 15, n. 2, p. 97–108, Fevereiro 1993.

[113] BONCELET, C. *Image Noise Models*. Alan C. Bovik. Handbook of Image and Video Processing: Academic Press, 2005.

[114] TAUBMAN, D. S.; MARCELLIN, M. W. *JPEG2000: Image Compression Fundamentals, Standards and Practice*. [S.l.]: Kluwer Academic, 2002.

[115] FERREIRA, R. et al. Video super-resolution based on local invariant features matching. In: *IEEE International Conference on Image Processing*. [S.l.: s.n.], 2012.