



**SUPER-RESOLUÇÃO BASEADA EM CORRESPONDÊNCIA
DE CARACTERÍSTICAS SIFT E CASAMENTO DE GRADIENTES**

RENAN UTIDA FERREIRA

**TESE DE DOUTORADO EM ENGENHARIA DE SISTEMAS ELETRÔNICOS E
AUTOMAÇÃO**

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

**FACULDADE DE TECNOLOGIA
UNIVERSIDADE DE BRASÍLIA**

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

**SUPER-RESOLUÇÃO BASEADA EM CORRESPONDÊNCIA
DE CARACTERÍSTICAS SIFT E CASAMENTO DE GRADIENTES**

RENAN UTIDA FERREIRA

ORIENTADOR: RICARDO LOPES DE QUEIROZ, PH.D

**TESE DE DOUTORADO ENGENHARIA DE SISTEMAS
ELETRÔNICOS E DE AUTOMAÇÃO**

PUBLICAÇÃO: PGEA.TD - 097/15

BRASÍLIA/DF: JULHO - 2015

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**SUPER-RESOLUÇÃO BASEADA EM CORRESPONDÊNCIA
DE CARACTERÍSTICAS SIFT E CASAMENTO DE GRADIENTES**

RENAN UTIDA FERREIRA

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 em engenharia de sistemas eletrônicos e de automação.

Banca Examinadora:

Ricardo Lopes de Queiroz, Ph.D
UnB/CIC (Orientador, *in absentia*)

Mylène Christine Queiroz de Farias, Ph.D
UnB/ENE (Presidente da Banca)

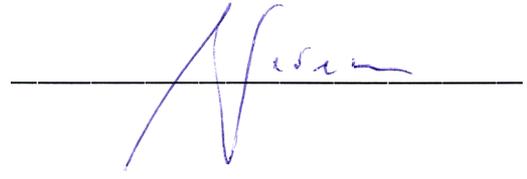
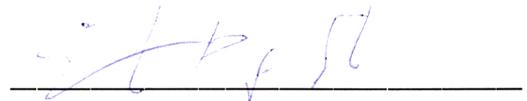
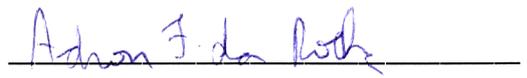
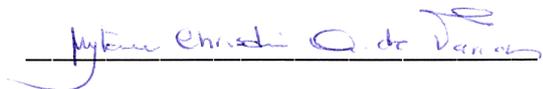
Adson Ferreira da Rocha, Ph.D
UnB/ENE (Examinador Interno)

Eduardo Peixoto Fernandes da Silva, Ph.D
UnB/ENE (Examinador Interno)

Bruno Luigi Macchiavello Espinoza, Dr.
UnB/CIC (Examinador Externo)

Fernando Manuel Bernardo Pereira, Ph.D
IST Lisboa (Examinador Externo)

Camilo Chang Dórea, Ph.D
UnB/CIC (Suplente)



BRASÍLIA, 31 DE JULHO DE 2015.

FICHA CATALOGRÁFICA

FERREIRA, RENAN UTIDA

Super-resolução Baseada em Correspondência de Características SIFT e Casamento de Gradientes [Distrito Federal] 2015.

xv, 116p., 210 x 297 mm (ENE/FT/UnB, Doutor, Engenharia de Sistemas Eletrônicos e Automação, 2015).

Tese de Doutorado – Universidade de Brasília, Faculdade de Tecnologia.

Departamento de Engenharia Elétrica

- | | |
|-----------------------------|----------------------------|
| 1. Processamento de imagens | 2. Processamento de vídeos |
| 3. Super-resolução | 4. Características SIFT |
| I. ENE/FT/UnB | II. Título (série) |

REFERÊNCIA BIBLIOGRÁFICA

FERREIRA, R.U. (2015). Super-resolução Baseada em Correspondência de Características SIFT e Casamento de Gradientes, Tese de Doutorado em Engenharia de Sistemas Eletrônicos e de Automação, Publicação PGEA.TD - 097/15, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 116p.

CESSÃO DE DIREITOS

AUTOR: Renan Utida Ferreira

TÍTULO: Super-resolução Baseada em Correspondência de Características SIFT e Casamento de Gradientes.

GRAU: Doutor ANO: 2015

É concedida à Universidade de Brasília permissão para reproduzir cópias desta tese de doutorado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desta tese de doutorado pode ser reproduzida sem autorização por escrito do autor.



Renan Utida Ferreira

Faculdade de Tecnologia

Departamento de Engenharia Elétrica (ENE)

Universidade de Brasília (UnB)

Campus Darcy Ribeiro

CEP 70910-900 - Brasília - DF - Brasil

*Aos meus pais, Clarice e Helber, e à
minha companheira, Victória.*

AGRADECIMENTOS

Primeiramente, agradeço aos meus pais, Clarice e Helber, por todo o apoio e incentivo ao longo de toda a minha vida. Agradeço também aos meus irmãos, Alice e Fabner, meus cunhados, Roberto e Myrella, meus sobrinhos, Gustavo e Marina, e minha irmã de coração, Andrea, que sempre estiveram ao meu lado me motivando a buscar meus sonhos. Agradeço aos meus professores que, em sua excelência acadêmica, se mostraram como modelos, em especial os professores Adson Ferreira da Rocha e Geovany Araújo Borges, por terem sido os primeiros a me guiarem na trilha da carreira acadêmica durante a graduação. Faço um agradecimento especial ao meu orientador de mestrado e doutorado, professor Ricardo Lopes de Queiroz, por ter acreditado em minha capacidade e ter me ajudado a me tornar o pesquisador que hoje sou. Agradeço aos companheiros de pesquisa, Alexandre, Bruno, Cauê, Diogo, Eduardo, Mintsu, Fabiano, Jorge, Rafael, Tiago, por tantas ideias compartilhadas. Pelos inúmeros momentos de descontração e diversão, agradeço aos queridos amigos, Bianca, Daniel Barbacena, Daniel Medeiros, Gabriel, Gracielle, Guilherme, João Marcos, Luís, Manuella, Marina, Mateus, Raquel e Sérgio. Agradeço à minha segunda família, Ermaine, João Victor, Celso, Antenor, Ulysses, por terem me recebido de braços abertos e termos compartilhado tantos bons momentos. Por fim, faço o agradecimento mais sincero e especial à minha companheira, Victória, por ter estado ao meu lado em todos os momentos, compartilhando desde minhas frustrações às minhas maiores conquistas. Seu amor e companheirismo foram imprescindíveis para a realização deste sonho.

RESUMO

SUPER-RESOLUÇÃO BASEADA EM CORRESPONDÊNCIA DE CARACTERÍSTICAS SIFT E CASAMENTO DE GRADIENTES

Autor: Renan Utida Ferreira

Orientador: Ricardo Lopes de Queiroz, Ph.D

Programa de Pós-graduação em Engenharia de Sistemas Eletrônicos e Automação

Brasília, 31 de julho de 2015

Reconstrução de imagem por super-resolução (ou simplesmente super-resolução) tem sido um campo de vasto estudo na área de processamento de imagens e vídeos. Dentre as várias propostas para resolver este problema, a super-resolução baseada em exemplos tem por objetivo transformar uma imagem em baixa resolução para uma imagem em alta resolução a partir da inferência de informações de alta frequência de outras imagens em alta resolução. Neste trabalho, propomos novas técnicas de super-resolução baseada em exemplos usando correspondência de descritores de características SIFT e casamento de gradientes. A correspondência de descritores é usada para a geração de um fluxo de vetores de movimentos, a partir dos quais compomos novas imagens por compensação de movimento usando transformações de perspectivas. O casamento de gradientes é usado para geração de imagens refinadas, a partir das quais será extraída informação de alta frequência a ser inserida na imagem de baixa resolução. Apresentamos dois métodos distintos para a solução proposta: no primeiro método, usamos grades móveis para selecionar os grupos de vetores que definem as transformações de perspectivas e casamento de gradientes em vizinhanças quadradas; no segundo, usamos agrupamento automático de vetores e casamento de gradientes em vizinhanças circulares. A solução proposta é testada no contexto de vídeos de resolução mista. Nossos resultados mostram qualidade objetiva dos quadros de vídeo super-resolvidos da ordem de 2dB, medida em PSNR, superior ao obtido por outras soluções.

ABSTRACT

SUPER-RESOLUTION BASED ON SIFT FEATURES CORRESPONDENCE AND GRADIENT MATCHING

Author: Renan Utida Ferreira

Advisor: Ricardo Lopes de Queiroz, Ph.D

Programa de Pós-graduação em Engenharia de Sistemas Eletrônicos e Automação

Brasília, 31th July 2015

Super-resolution image reconstruction (or simply super-resolution) has been a vast field of study in the area of image and video processing. Among the many solutions for this problem, example-based super-resolution has the objective of transforming a low resolution image into a high resolution one by inferring high frequency information from other high resolution images. In this work, we propose new techniques of example-based super-resolution by using SIFT feature descriptors matching followed by gradient matching. The descriptors' matching is used to generate a motion vector flow, from which we compose new images by motion compensation using perspective transformation. The gradient matching is used to generate refined images, from which we extract the high frequency information to be inserted to the low resolution image. We present two distinct methods for this solution: in the first method, we use moving grids to select the groups of vectors from which we derive the perspective transformation, followed by the matching of gradients in a square neighborhood; in the second method, we use the automatic clustering of vectors and gradient matching in a circular neighborhood. The proposed solution is tested in the context of mixed resolution videos. Our results show superior objective quality in the super-resolved video frames of around 2dB, measured in PSNR, compared to other solutions.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	CONTEXTUALIZAÇÃO	1
1.2	DEFINIÇÃO DO PROBLEMA	1
1.3	OBJETIVOS	2
1.4	ORGANIZAÇÃO DO DOCUMENTO	2
2	FUNDAMENTAÇÃO TEÓRICA	5
2.1	INTRODUÇÃO	5
2.2	CONCEITOS DE IMAGEM E VÍDEO DIGITAIS	5
2.2.1	GRADIENTE DE IMAGEM	6
2.2.2	ESPAÇO DE CORES	7
2.2.3	RESOLUÇÃO	9
2.3	REDIMENSIONAMENTO DE IMAGEM	10
2.3.1	INTERPOLAÇÃO	10
2.3.2	DECIMAÇÃO	12
2.3.3	RELAÇÃO ENTRE INTERPOLAÇÃO E DECIMAÇÃO	13
2.4	SUPER-RESOLUÇÃO	14
2.4.1	MODELO DE IMAGEAMENTO	15
2.4.2	SUPER-RESOLUÇÃO POR INTERPOLAÇÃO-RESTAURAÇÃO	16
2.4.3	SUPER-RESOLUÇÃO POR ABORDAGEM ESTOCÁSTICA	17
2.4.4	SUPER-RESOLUÇÃO POR PROJEÇÃO EM CONJUNTOS CONVEXOS	18
2.4.5	SUPER-RESOLUÇÃO NO DOMÍNIO DA FREQUÊNCIA	19
2.4.6	SUPER-RESOLUÇÃO BASEADA EM EXEMPLOS	20
2.4.7	SUPER-RESOLUÇÃO DE VÍDEO	21
2.4.7.1	SUPER-RESOLUÇÃO DE VÍDEO DE RESOLUÇÃO MISTA POR PONDERAÇÃO DE DICIONÁRIO	23
2.5	AVALIAÇÃO DE QUALIDADE DE IMAGEM E VÍDEO	24
2.6	CARACTERÍSTICAS E DESCRITORES	26
2.6.1	CARACTERÍSTICAS INVARIANTES LOCAIS	26
2.6.2	DESCRITORES NÃO-BINÁRIOS	29
2.6.2.1	SIFT	29
2.6.2.2	SURF	31
2.6.3	DESCRITORES BINÁRIOS	32
2.7	GEOMETRIA PROJETIVA 2D E TRANSFORMAÇÕES BIDIMENSIONAIS	34
2.7.1	CLASSES DE HOMOGRAFIA	36
2.7.2	DEFININDO UMA HOMOGRAFIA A PARTIR DE PONTOS DE IMAGENS	38

3	FORMULAÇÃO DO PROBLEMA E ESTRUTURA GERAL DA SOLUÇÃO PROPOSTA	41
3.1	COMPENSAÇÃO DE IMAGEM BASEADA EM CORRESPONDÊNCIA DE CARACTERÍSTICAS SIFT.....	45
3.2	CONSTRUÇÃO DE IMAGEM POR CASAMENTO DE GRADIENTES	50
3.3	COMBINANDO COMPENSAÇÃO POR CORRESPONDÊNCIA DE CARACTERÍSTICAS E CASAMENTO DE GRADIENTES PARA SUPER-RESOLUÇÃO	51
4	COMPENSAÇÃO DE MOVIMENTO BASEADA EM GRADES MÓVEIS	53
4.1	COMPENSAÇÃO DE MOVIMENTO BASEADA EM GRADES MÓVEIS	53
4.2	CASAMENTO DE GRADIENTES EM VIZINHANÇAS QUADRADAS	57
4.3	SUPER-RESOLUÇÃO DE QUADROS DE VÍDEO	61
4.4	ADIÇÃO DIRETA DE INFORMAÇÃO DE ALTA FREQUÊNCIA	64
4.5	SUPER-RESOLUÇÃO POR ANÁLISE ESTATÍSTICA DA INFORMAÇÃO DE ALTA FREQUÊNCIA	70
4.6	SUPER-RESOLUÇÃO POR PONDERAÇÃO DE DICIONÁRIO	74
4.7	SUPER-RESOLUÇÃO POR PONDERAÇÃO DE DICIONÁRIO COMBINADO	77
4.8	COMPARAÇÃO DE RESULTADOS	80
5	COMPENSAÇÃO DE MOVIMENTO BASEADA EM AGRUPAMENTO AUTOMÁTICO DE VETORES	85
5.1	COMPENSAÇÃO DE MOVIMENTO BASEADA EM AGRUPAMENTO DE VETORES	85
5.1.1	AGRUPAMENTO DE VETORES	86
5.1.2	DETERMINAÇÃO DAS REGIÕES DE RECORTE	87
5.1.3	COMPENSAÇÃO DE IMAGEM PARA DIFERENTES NÚMEROS DE GRUPOS DE VETORES	90
5.2	CASAMENTO DE GRADIENTES EM VIZINHANÇAS CIRCULARES	92
5.3	SUPER-RESOLUÇÃO DE QUADROS DE VÍDEO	95
5.3.1	CONDIÇÕES DE TESTE.....	95
5.3.2	RESULTADOS EXPERIMENTAIS	97
5.3.3	ANÁLISE DOS RESULTADOS	100
5.4	SUPER-RESOLUÇÃO DE IMAGENS SOB TRANSFORMAÇÕES DIVERSAS	101
5.4.1	ANÁLISE DOS RESULTADOS	104
6	CONCLUSÃO	107
6.1	COMPENSAÇÃO DE MOVIMENTO BASEADA EM GRADES MÓVEIS	107
6.2	COMPENSAÇÃO DE MOVIMENTO BASEADA EM AGRUPAMENTO DE VETORES	108
6.3	CONSIDERAÇÕES FINAIS E TRABALHOS TUTUROS	109
	REFERÊNCIAS BIBLIOGRÁFICAS	110

LISTA DE FIGURAS

2.1	Representação de uma sequência de quadros de vídeo.	6
2.2	Exemplo de (a) imagem original; (b) magnitude do vetor gradiente; (c) derivada parcial na direção x (vertical); (d) derivada parcial na direção y (horizontal).	7
2.3	Exemplo de uma imagem colorida e suas componentes de cores: (a) imagem colorida com as três componentes; (b) componente R (vermelho); (c) componente G (verde); (d) componente B (azul).	8
2.4	Exemplo de componentes no espaço YCbCr: (a) componente Y (Luminância); (c) componente Cb (crominância, diferença do azul); (d) componente Cr (crominância, diferença do vermelho).	8
2.5	Exemplo de um bloco de <i>pixels</i> interpolado por um fator de escala $E = 2$ e filtrado com filtro bilinear.	11
2.6	Exemplo de interpolação com <i>zoom</i> : (a) imagem original; (b) interpolação com vizinho mais próximo; (c) interpolação com filtro bilinear; (d) interpolação com filtro bicúbico; (e) interpolação com filtro lanczos3.	12
2.7	Exemplo de decimação: (a) imagem original; (b) imagem decimada sem filtro <i>anti-aliasing</i> ; (c) imagem decimada com filtro <i>anti-aliasing</i> bicúbico.	13
2.8	Exemplo de decimação com <i>zoom</i> : (a) imagem original; (b) imagem decimada sem filtro <i>anti-aliasing</i> ; (c) imagem decimada com filtro <i>anti-aliasing</i> bicúbico.	14
2.9	Reamostragem como sequência dos processos de subamostragem e subamostragem... ..	14
2.10	Exemplo de reamostragem com diferentes combinações de filtros: (a) imagem original; (b) filtro bilinear para subamostragem e sobreamostragem; (c) filtro lanczos3 para subamostragem e bilinear para sobreamostragem; (d) filtro bilinear para subamostragem e lanczos3 para sobreamostragem; (e) filtro lanczos3 para subamostragem e sobreamostragem.	15
2.11	SR por interpolação baseada em alinhamento e “desborramento”.	17
2.12	Exemplo de SR baseada em exemplos: (a) imagem em BR; (b) imagem interpolada; (c) imagem super-resolvida; (d) imagem original em AR; (e-g) imagens de referência em AR; (h) recortes da imagem super-resolvida; (i) recortes das imagens de referência usados para obter os recortes super-resolvidos em (h)	22
2.13	Exemplo de uma sequência de vídeo em resolução mista com quadro-chave na resolução original e quadros-não-chave com resolução e tamanho reduzidos.	23
2.14	Detecção de pontos extremos.	29
2.15	Gradientes e descritores.	31
2.16	Da esquerda para a direita: Derivadas parciais de segunda ordem da Gaussiana nas direções y e xy , e a aproximação das derivadas por filtros do tipo <i>box</i>	32
2.17	Exemplos de padrões de amostragem de pontos para descritores binários: (a) BRIEF; (b) ORB; (c) BRISK; (d) FREAK; (e) RIFF.	34

2.18	Modelo do plano projetivo	35
2.19	Mapeamento entre os pontos de dois planos.	36
2.20	Exemplo de mapeamento pela matriz de homografia H de um quadrado entre dois sistemas de coordenadas representados pelos planos π_1 e π_2 , pela matriz de homografia H , em que são mostrados os pontos de fuga.	38
2.21	Exemplo de transformação de perspectiva	39
3.1	Exemplo de descritores SIFT sobrepostos a imagens em (a) baixa resolução e (b) alta resolução.	43
3.2	Correspondência dos descritores mostrados na figura 3.1.	44
3.3	Diagrama geral da compensação de movimento baseada em correspondência de características.	46
3.4	Exemplo de vetores de correspondência sobrepostos à imagem Org^{baixa}	47
3.5	Exemplo de agrupamento do fluxo de vetores em três grupos e bordas das regiões de recorte.	48
3.6	Exemplo de composição da imagem $Comp$ a partir das máscaras binárias e versões distorcidas da imagem Ref	49
3.7	Exemplo de imagem $Comp$ resultante da compensação de movimento.	49
3.8	Diagrama de operações para a composição de imagem por casamento de gradientes.	51
3.9	Exemplo de casamento de gradientes, considerando uma vizinhança do $pixel$. ..	51
3.10	Diagrama descrevendo a combinação de técnicas aplicada a super-resolução. ...	52
4.1	Compensação de movimento baseada em grades móveis.	54
4.2	Exemplo de compensação: (a) imagem em baixa resolução interpolada Org^{baixa} com grade sobreposta; (b) imagem em alta resolução Ref ; (c) imagem compensada $Comp$. Imagens transformadas: (d) $\tau_1(Ref)$; (e) $\tau_2(Ref)$; (f) $\tau_3(Ref)$; (g) $\tau_4(Ref)$	55
4.3	Exemplos de grades com $TGrade = 64 pixels$ e (a) $DGrade = (0, 0)$, (b) $DGrade = (32, 32)$; (c) $TGrade = 96 pixels$ e $DGrade = (0, 0)$. Os deslocamentos, em $pixels$, são dados nas direções horizontal e vertical, respectivamente.	55
4.4	Exemplo do conjunto $\{Comp(k)\}$ usando $TGrade = 128 pixels$: (a) imagem Org^{baixa} ; (b) imagem Ref ; imagem Org^{baixa} com grades (c) $k = 1$ e (d) $k = 2$ sobreposta; (e) $Comp(1)$; (f) $Comp(2)$; (g) conjunto de imagens $\{Comp(k)\}$	56
4.5	Exemplo de imagens aprimoradas após o casamento de gradientes para $TVIz = 0$: (a) $Aper(0)$, (b) $Aper^{baixa}(0)$ e (c) $Bord(0)$	58
4.6	Exemplo de imagens aprimoradas após o casamento de gradientes para $TVIz = 7$: (a) $Aper(7)$, (b) $Aper^{baixa}(7)$ e (c) $Bord(7)$	59
4.7	Exemplo de imagens aprimoradas após o casamento de gradientes para $TVIz = 15$: (a) $Aper(15)$, (b) $Aper^{baixa}(15)$ e (c) $Bord(15)$	60

4.8	Quadros da sequência <i>container</i> usados nos testes: (a) 1° quadro original, (b) 16° quadro reamostrado e (c) 31° quadro original.....	62
4.9	Quadros da sequência <i>hall</i> usados nos testes: (a) 1° quadro original, (b) 16° quadro reamostrado e (c) 31° quadro original.....	62
4.10	Quadros da sequência <i>mobile</i> usados nos testes: (a) 1° quadro original, (b) 16° quadro reamostrado e (c) 31° quadro original.....	62
4.11	Quadros da sequência <i>news</i> usados nos testes: (a) 1° quadro original, (b) 16° quadro reamostrado e (c) 31° quadro original.....	63
4.12	Quadros da sequência <i>mobcal</i> usados nos testes: (a) 1° quadro original, (b) 16° quadro reamostrado e (c) 31° quadro original.....	63
4.13	Quadros da sequência <i>shields</i> usados nos testes: (a) 1° quadro original, (b) 16° quadro reamostrado e (c) 31° quadro original.....	63
4.14	Comparação de valores de PSNR entre diferentes usos de quadros-chave, variando TViz para sequência <i>container</i> e tamanhos de blocos: (a) Pequenos, (b) Médios e (c) Grandes.....	65
4.15	Comparação de valores de PSNR entre diferentes usos de quadros-chave, variando TViz para sequência <i>hall</i> e tamanhos de blocos: (a) Pequenos, (b) Médios e (c) Grandes.....	65
4.16	Comparação de valores de PSNR entre diferentes usos de quadros-chave, variando TViz para sequência <i>mobile</i> e tamanhos de blocos: (a) Pequenos, (b) Médios e (c) Grandes.....	65
4.17	Comparação de valores de PSNR entre diferentes usos de quadros-chave, variando TViz para sequência <i>news</i> e tamanhos de blocos: (a) Pequenos, (b) Médios e (c) Grandes.....	66
4.18	Comparação de valores de PSNR entre diferentes usos de quadros-chave, variando TViz para sequência <i>mobcal</i> e tamanhos de blocos: (a) Pequenos, (b) Médios e (c) Grandes.....	66
4.19	Comparação de valores de PSNR entre diferentes usos de quadros-chave, variando TViz para sequência <i>shields</i> e tamanhos de blocos: (a) Pequenos, (b) Médios e (c) Grandes.....	66
4.20	Curvas de PSNR dos gráficos mostrados nas Figuras 4.14a a 4.19c normalizadas, onde buscamos observar qual valor de $TViz$ concentra mais pontos próximos a 1.	68
4.21	Histogramas dos valores de PSNR normalizados mostrados nas Figuras 4.20, para cada valor de $TViz$, referentes à SR por adição direta.....	69
4.22	Geração da imagem com informação de alta frequência $Bord^{AE}$ a partir da média dos <i>pixels</i> colocalizados das imagens do conjunto <i>Bord</i> composto pela união de todos os conjuntos $\{Bord_{TGrade}(v)\}_n$	71
4.23	Comparação de valores de PSNR resultantes da composição de conjuntos para diferentes valores de tamanho da vizinhança $TViz$, para todas as sequências: (a) <i>container</i> ; (b) <i>hall</i> ; (c) <i>mobile</i> ; (d) <i>news</i> ; (e) <i>mobcal</i> ; (f) <i>shields</i>	72

4.24	Curvas de PSNR dos gráficos mostrados nas Figuras 4.23a a 4.23f normalizadas.....	72
4.25	Histograma dos valores de PSNR normalizados mostrados na Figura 4.24, para cada valor de $TViz$, referentes à SR por análise estatística.....	73
4.26	Comparação de valores de PSNR resultantes da composição de conjuntos para diferentes tamanhos de dicionário (representados pelo tamanho da vizinhança $TViz$), para todas as sequências: (a) <i>container</i> ; (b) <i>hall</i> ; (c) <i>mobile</i> ; (d) <i>news</i> ; (e) <i>mobcal</i> ; (f) <i>shields</i>	75
4.27	Curvas de PSNR dos gráficos mostrados nas Figuras 4.26a a 4.26f para blocos de tamanho 16×16 , normalizadas.	76
4.28	Histograma dos valores de PSNR normalizados mostrados na Figura 4.27, para cada valor de $TViz$, referentes à SR por ponderação de dicionário.	76
4.29	Comparação de valores de PSNR resultantes da composição de conjuntos para diferentes tamanhos de dicionário (representados pelo tamanho da vizinhança $TViz$), para todas as sequências: (a) <i>container</i> ; (b) <i>hall</i> ; (c) <i>mobile</i> ; (d) <i>news</i> ; (e) <i>mobcal</i> ; (f) <i>shields</i>	78
4.30	Curvas de PSNR dos gráficos mostrados nas Figuras 4.29a a 4.29f normalizadas, para blocos de tamanho 8×8 para sequências CIF e tamanho 16×16 para sequências 720p.	78
4.31	Histograma dos valores de PSNR normalizados mostrados na Figura 4.30, para cada valor de $TViz$, referentes à SR por ponderação de dicionário conjunto com OBMC... ..	79
4.32	Exemplo comparativo visual para a sequência <i>news</i> , com <i>zoom</i> : (a) quadro original; (b) quadro interpolado com filtro Lanczos-3; (c) quadro super-resolvido por adição direta; (d) quadro super-resolvido por ponderação de dicionário conjunto com SIFT e OBMC.	82
4.33	Exemplo comparativo visual para a sequência <i>mobile</i> , com <i>zoom</i> : (a) quadro original; (b) quadro interpolado com filtro Lanczos-3; (c) quadro super-resolvido por adição direta; (d) quadro super-resolvido por ponderação de dicionário conjunto com SIFT e OBMC.....	83
5.1	Diagrama da compensação de movimento baseada em agrupamento de vetores.	86
5.2	Exemplo de fluxo de vetores e seu agrupamento: (a) fluxo não agrupado; (b) fluxo completo separado em três grupos.....	87
5.3	Exemplo de detecção e remoção de vetores discrepantes dentro de cada grupo: (a) vetores discrepantes destacados; (b) vetores discrepantes removidos.	88
5.4	Exemplo de definição das Regiões de Interesse (RdI): (a) feixo convexo; (b) regiões internas aos feixos convexas não-sobrepostas; (c) regiões segmentadas por <i>watershed</i> . ..	89
5.5	Exemplo de imagens compensadas com as bordas das RdI sobrepostas: (a) <i>Comp</i> (3); (b) <i>Comp</i> ^{baixa} (3).	91
5.6	Exemplo de obtenção de v_{max} : (a) quadro-não-chave com vetores separados em $NGrupos = 18$, com feixos convexas e bordas de RdIs; (b) todas as RdI; (c) círculo e elemento estruturante octogonal com $v_{max} = 17$	94

5.7	Exemplo comparativo visual para a sequência <i>mobile</i> , com <i>zoom</i> : (a) quadro original; (b) ISR; (c) DSR; (d) PDO-SR (MAV).	99
5.8	Imagens do banco <i>bark</i> usadas nos testes: (a) 1ª imagem original, (b) 3ª imagem reescalada (decimado e interpolado) e (c) 6ª imagem original.....	101
5.9	Imagens do banco <i>bikes</i> usadas nos testes: (a) 1ª imagem original, (b) 3ª imagem reescalada (decimado e interpolado) e (c) 6ª imagem original.....	102
5.10	Imagens do banco <i>boat</i> usadas nos testes: (a) 1ª imagem original, (b) 3ª imagem reescalada (decimado e interpolado) e (c) 6ª imagem original.....	102
5.11	Imagens do banco <i>graf</i> usadas nos testes: (a) 1ª imagem original, (b) 3ª imagem reescalada (decimado e interpolado) e (c) 6ª imagem original.....	102
5.12	Imagens do banco <i>leuven</i> usadas nos testes: (a) 1ª imagem original, (b) 3ª imagem reescalada (decimado e interpolado) e (c) 6ª imagem original.....	102
5.13	Imagens do banco <i>trees</i> usadas nos testes: (a) 1ª imagem original, (b) 3ª imagem reescalada (decimado e interpolado) e (c) 6ª imagem original.....	103
5.14	Imagens do banco <i>ubc</i> usadas nos testes: (a) 1ª imagem original, (b) 3ª imagem reescalada (decimado e interpolado) e (c) 6ª imagem original.....	103
5.15	Imagens do banco <i>wall</i> usadas nos testes: (a) 1ª imagem original, (b) 3ª imagem reescalada (decimado e interpolado) e (c) 6ª imagem original.....	103

LISTA DE TABELAS

4.1	Diferença de PSNR [dB] entre o uso de dois e um quadros-chave como referência	67
4.2	Comparação de valores de PSNR para as diferentes técnicas de SR.....	80
5.1	Valores de PSNR médios para diferentes combinações de filtros de reamostragem para todos os quadros das sequências: (a) <i>container</i> ; (b) <i>hall</i> ; (c) <i>mobile</i> ; (d) <i>news</i> ; (e) <i>mobcal</i> ; (f) <i>shields</i>	96
5.2	Comparação de valores de PSNR para diferentes técnicas, sob Cenário 1	97
5.3	Comparação de valores de PSNR para diferentes técnicas, sob Cenário 2	98
5.4	Comparação de valores de PSNR para diferentes técnicas, sob Cenário 1	104
5.5	Comparação de valores de PSNR para diferentes técnicas, sob Cenário 2	105

LISTA DE ABREVIACOES, ACRONIMOS E SMBOLOS

AD-SR	Super-resoluo por adio direta
AE-SR	Super-resoluo por anlise estatstica
AGAST	Teste de segmento acelerado adaptativo e genrico (do ingls <i>Adaptive and Generic Accelerated Segment Test</i>)
AR	Alta resoluo
BR	Baixa resoluo
BRIEF	Descriptor das caractersticas elementares independentes robustas e binrias (do ingls <i>Binary robust independent elementary features</i>)
BRISK	Pontos de interesse escalonveis invariantes robustos e binrios <i>Binary robust invariant scalable keypoints</i>)
CFT	Transformada contnua de fourier (do ingls <i>continuous fourier transform</i>)
CIELAB	Espao de cores
CIF	Formato intermedirio comum (do ingls <i>Common intermediate format</i>)
CMY(K)	Espao de cores
DFT	Transformada discreta de fourier (do ingls <i>discrete fourier transform</i>)
DGrade	Deslocamento da grade
DLT	Transformao linear direta (do ingls <i>Direct Linear Transformation</i>)
DoG	Diferena de Gaussianas (do ingls <i>Difference of Gaussians</i>)
FAST	Caractersticas de teste de segmento acelerado (do ingls <i>Features from accelerated segment test</i>)
FREAK	Ponto de interesse de retina rpido (do ingls <i>Fast Retina Keypoint</i>)
GLOH	Histograma de localizao e orientao de gradiente (do ingls <i>Gradient location and orientation histogram</i>)
GPU	Unidade de processamento grfico (do ingls <i>Graphic processing unit</i>)
HD	Alta definio (do ingls <i>High definition</i>)
HSI	Espao de cores
HSR	Super-resoluo hbrida
ISR	Super-resoluo de imagem nica
LoG	Laplaciana da Guassiana (do ingls <i>Laplacian of Guassian</i>)
MAV	Mtodo de agrupamento de vetores
MGM	Mtodo de grades mveis
MSE	Erro mdio quadrtico (do ingls <i>Mean squared error</i>)

MSR	Super-resolução por compensação de movimento
NGrupos	Número de grupos de vetores
OBMC	Compensação de movimento com sobreposição de bloco (do inglês <i>Overlapped-block motion compensation</i>)
ORB	FAST orientado e BRIEF rotacionado (do inglês <i>Oriented FAST and rotated BRIEF</i>)
PCA	Análise de componente principal (do inglês <i>Principal component analysis</i>)
PCS	Projeção em conjuntos convexos (do inglês <i>Projections onto convex sets</i>)
PD-SR	Super-resolução por ponderação de dicionário
PDO-SR	Super-resolução por ponderação de dicionário com OBMC
PSNR	Relação sinal-ruído de pico (do inglês <i>Peak signal-to-noise ratio</i>)
RANSAC	consenso de amostras aleatórias (do inglês <i>random sample consensus</i>)
RdI	Região de interesse
RGB	Espaço de cores
RIFF	Descritor de característica rápido e invariante inspirado em retina (do inglês <i>retina-inspired invariant fast feature descriptor</i>)
SAD	Soma das diferenças absolutas (do inglês <i>Sum of absolute differences</i>)
SIFT	Transformada de feições invariantes a escala (do inglês <i>Scale invariant feature transform</i>)
SR	Super-resolução
SSD	Soma de diferenças quadráticas (do inglês <i>Sum of squared differences</i>)
SURF	Características robustas aceleradas (do inglês, <i>speeded-Up robust features</i>)
TGrade	Tamanho de regiões de grade
TViz	Tamanho de vizinhança
YCbCr	Espaço de cores

Capítulo 1

Introdução

1.1 CONTEXTUALIZAÇÃO

As tecnologias de captura e visualização de imagens digitais têm avançado bastante nas duas últimas décadas. Temos notado o aumento na riqueza de detalhes em imagens tanto pelos dispositivos de captura, como câmeras, quanto em sua visualização por meio de aparelhos como televisores e projetores. Este aumento vai ao encontro do grande interesse que se tem (em geral) em aplicações de imagens digitais por imagens de alta resolução (AR). Essas imagens trazem em si uma riqueza maior de detalhes se comparadas àquelas de baixa resolução (BR), riqueza esta desejada especialmente em duas grandes áreas de aplicação: na melhora de informação pictórica para interpretação humana e na ajuda na representação para automação na percepção por máquinas [1]. Claramente, os termos alta e baixa resolução são relativos e dependem de um contexto, sempre intimamente ligado a uma maior ou menor riqueza de detalhes.

Apesar do aumento na resolução de captura, pesquisas têm sido desenvolvidas há bastante tempo [2] buscando o aumento de resolução de imagens já capturadas para as mais diversas aplicações como imagens médicas, imageamento por satélite, leitura de placas de trânsito, melhora de imagens e vídeos comprimidos, entre outras [3]. Este processo de aumento de resolução de uma imagem é conhecido por super-resolução (SR) e se dá, de forma geral, pela obtenção de uma imagem em alta resolução a partir de uma ou mais imagens em baixa resolução [4]. As técnicas de SR permitem que se obtenham imagens de resolução mais alta do que aquela limitada pelos sistemas de captura. Uma outra forma de se super-resolver uma imagem se dá por tomar os detalhes de uma imagem em AR e aplicá-los a uma imagem em BR, conhecida com SR baseada em exemplos [5].

1.2 DEFINIÇÃO DO PROBLEMA

Inúmeros trabalhos têm sido desenvolvidos no campo de super-resolução baseada em exemplos. Contudo, boa parte deles é baseada na correlação entre os *pixels* da imagem. Es-

sas abordagens têm uma certa limitação quando observamos as diferenças entre a imagem em baixa-resolução a ser super-resolvida e as imagens em alta-resolução usadas como referência, das quais se obtêm informações sobre os detalhes da imagem. Essas limitações estão principalmente relacionadas a variações de iluminação, perspectiva de observação da cena, movimentação de objetos em relação à câmera, entre outros.

Pode-se buscar ir além dessas limitações a partir da semelhança entre características presentes nas imagens e não somente usando da correlação entre *pixels*. O uso de características permite então que se suponha uma premissa para quais tipos de imagens de referência usar. Assim, de forma a garantir que características mais sofisticadas sejam compartilhadas entre as imagens em baixa e alta resoluções, pode-se tomar a premissa de que todas as imagens usadas tenham sido capturadas de uma mesma cena. Neste contexto, as sequências de vídeo, por conterem quadros bastante correlacionados, são as coleções de imagens onde mais facilmente se tem a premissa assegurada.

Entre outras aplicações, duas podem ser alvo desta abordagem de super-resolução. A primeira é no caso de vídeos de resolução mista, ou seja, vídeos compostos por quadros tanto em alta quanto em baixa resolução. A segunda é no caso de vídeos com simultâneas capturas de imagens em resolução superior àquela do vídeo.

1.3 OBJETIVOS

Este trabalho tem por objetivo geral a super-resolução baseada em exemplos de quadros de vídeo e imagens. Como objetivos específicos, buscamos:

- Aproveitar redundância de informação de imagens capturadas de uma mesma cena;
- Usar correspondência de características para estimação de movimentos mais complexos que translação;
- Superar técnicas baseadas em semelhanças de *pixels*.

Foram realizados testes usando sequências de vídeos não-comprimidos, bem como bancos de imagens capturadas de uma mesma cena. Todas as imagens e vídeos consistem em cenas naturais, ou seja, não incluem imagens sintéticas, como animações, imagens de ressonância magnética, etc. Todos os testes foram executados considerando apenas as informações de luminância das imagens.

1.4 ORGANIZAÇÃO DO DOCUMENTO

O Capítulo 2 desta tese traz a fundamentação teórica necessária para o bom entendimento do trabalho, incluindo uma revisão bibliográfica. O Capítulo 3 apresenta e descreve a solução proposta. O Capítulo 4 traz a implementação e os respectivos testes da solução usando um

método de grades móveis, com resultados preliminares apresentados em [6], já mostrando superioridade da solução proposta sobre técnicas baseadas em semelhança entre *pixels*. O Capítulo 5 traz um método melhorado, baseado em agrupamento de vetores de características para a decisão automática de parâmetros, com resultados apresentados em [7], mostrando desempenho superior ao método anterior. O Capítulo 6 traz as conclusões do trabalho.

Capítulo 2

Fundamentação Teórica

2.1 INTRODUÇÃO

Neste capítulo, trazemos informações importantes para o entendimento geral do trabalho proposto. Apresentamos também uma revisão bibliográfica acerca dos conceitos mais relevantes, além de técnicas relacionadas ao problema de super-resolução. Primeiro apresentamos conceitos relacionados a imagens digitais, como gradiente de imagens, definições de tipos de resolução e operações de mudança de tamanho de imagem por decimação e interpolação. Em seguida, mostramos alguns diferentes tipos de super-resolução de imagem e vídeo. Como o método que propomos é baseado em correspondência de características, fazemos uma revisão de detectores e descritores de características. Os pares de características correspondidas são usados para derivar transformações entre imagens e por isso trazemos, ao final do capítulo, uma breve introdução teórica acerca de transformações projetivas.

2.2 CONCEITOS DE IMAGEM E VÍDEO DIGITAIS

Uma imagem digital é uma representação discreta de um sinal bidimensional. Matematicamente, ela pode ser representada como uma função bidimensional $f(x, y)$, na qual x e y assumem valores discretos não negativos, podendo ser expressa como uma matriz de tamanho $M \times N$

$$f(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & \cdots & f(0, N - 1) \\ f(1, 0) & f(1, 1) & \cdots & f(1, N - 1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M - 1, 0) & f(M - 1, 1) & \cdots & f(M - 1, N - 1) \end{bmatrix} \quad (2.1)$$

Também é comum representar uma imagem usando a notação de matriz bidimensional F cujas entradas $F_{i,j}$ (ou $f(x, y)$), representam os elementos da imagem, ou *pixels* (do inglês, *picture element*). Uma outra forma de representar uma imagem é pela sua escrita como um vetor em notação lexicográfica $\mathbf{f} = [f(0, 0), \dots, f(0, N - 1), \dots, f(M - 1, N - 1)]^T$.

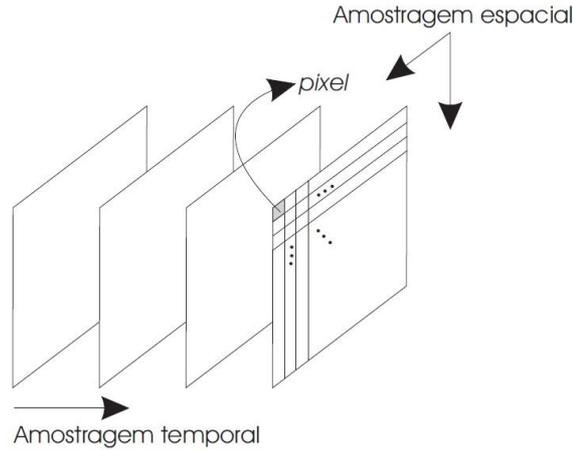


Figura 2.1: Representação de uma sequência de quadros de vídeo.

Um vídeo digital é uma sequência de imagens digitais correlacionadas temporalmente, conforme ilustrado na Figura 2.2, ou seja, é um sinal tridimensional discretizado em duas dimensões espaciais e uma temporal. Matematicamente, um vídeo pode ser representado por uma matriz tridimensional \mathbf{V} cujas entradas são $v(x, y, t)$, em que t representa a dimensão temporal. Cada imagem na sequência de vídeo é também conhecida com “quadro”.

2.2.1 Gradiente de imagem

Em Cálculo Vetorial, para uma função contínua bidimensional $f(x, y)$ com derivadas parciais f_x e f_y , o operador gradiente (∇) desta função é definido por

$$\nabla f(x, y) = \begin{bmatrix} \frac{\partial}{\partial x} f(x, y) \\ \frac{\partial}{\partial y} f(x, y) \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix} \quad (2.2)$$

Para uma função discreta unidimensional $f(x)$, a sua derivada pode ser definida como a diferença $f(x + 1) - f(x)$. Contudo, para funções bidimensionais, a derivada pode ser aproximada de diversas formas, sendo uma das mais conhecidas o uso dos operadores de Sobel. Estes operadores são definidos pelas matrizes \mathbf{S}_x e \mathbf{S}_y abaixo. De posse desses operadores, as derivadas parciais são calculadas por $f_x = f(x, y) * \mathbf{S}_x(x, y)$ e $f_y = f(x, y) * \mathbf{S}_y(x, y)$, onde $*$ representa o operador de convolução bidimensional [8].

$$\mathbf{S}_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \mathbf{S}_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Como o gradiente é um vetor, existe um interesse em sua magnitude, calculada por $m(x, y) = \text{mag}(\nabla f) = \sqrt{f_x^2 + f_y^2}$, e seu ângulo, calculado por $\theta(x, y) = \text{ang}(\nabla f) = \tan^{-1}(f_y/f_x)$. A Figura 2.2 mostra exemplos de cada uma das derivadas parciais e da mag-

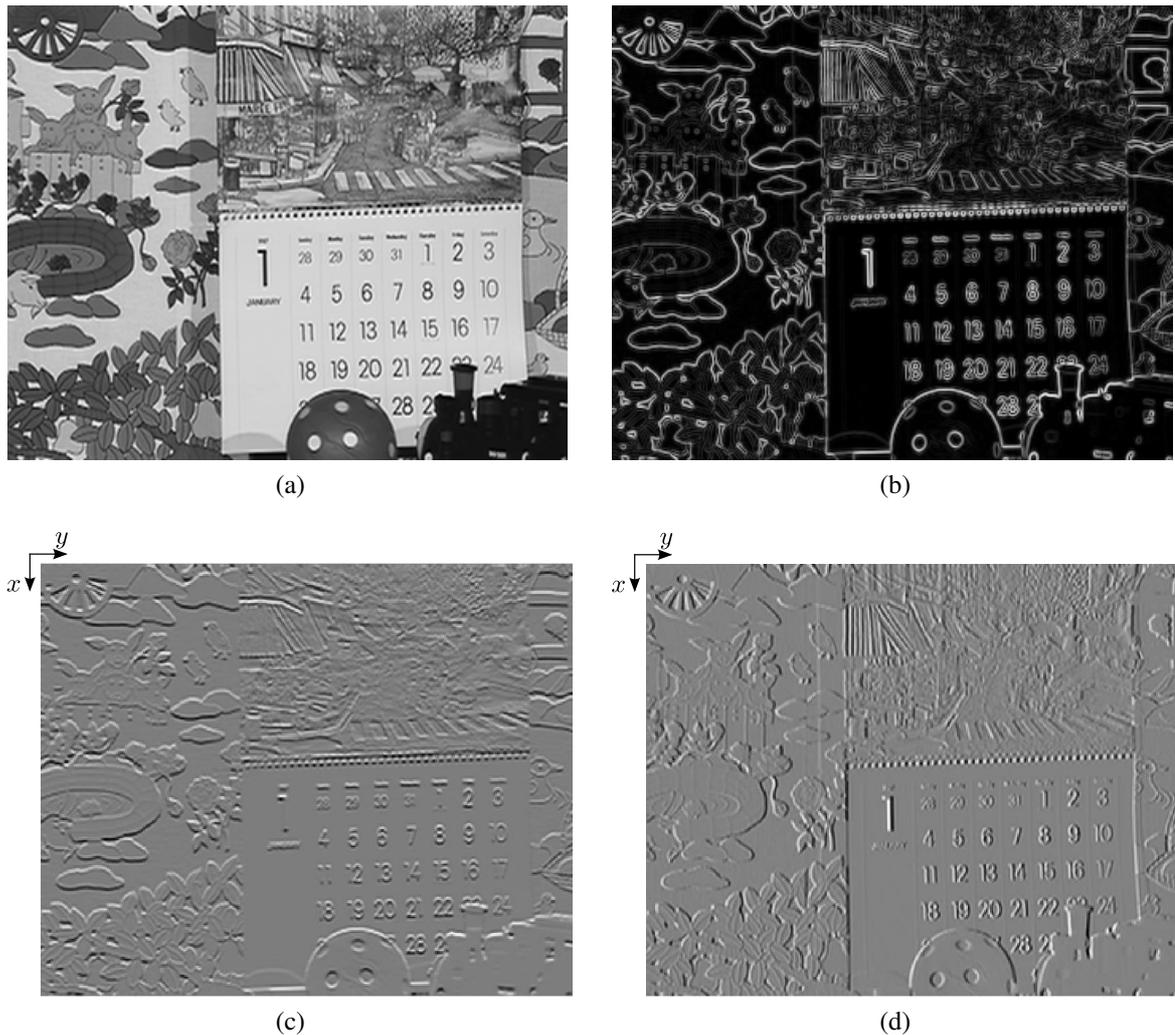


Figura 2.2: Exemplo de (a) imagem original; (b) magnitude do vetor gradiente; (c) derivada parcial na direção x (vertical); (d) derivada parcial na direção y (horizontal).

nitidade do gradiente.

Como se pode notar, a magnitude do vetor gradiente é capaz de indicar fortes variações entre *pixels* vizinhos. Isto é bastante útil para a obtenção de contornos da imagem.

2.2.2 Espaço de cores

Uma imagem em escala de cinza é representada por apenas uma matriz bidimensional, cujas entradas representam a intensidade dos *pixels*. Já uma imagem colorida é representada por mais de uma matriz bidimensional ou por uma matriz tridimensional. A representação de cores mais comum se dá pelo uso do espaço de cores *RGB*, onde se tem três matrizes bidimensionais contendo as componentes de cores vermelha (do inglês *Red*), verde (do inglês *Green*) e azul (do inglês *Blue*). A Figura 2.3 mostra um exemplo de uma imagem colorida e suas três matrizes R , G e B .



Figura 2.3: Exemplo de uma imagem colorida e suas componentes de cores: (a) imagem colorida com as três componentes; (b) componente R (vermelho); (c) componente G (verde); (d) componente B (azul).



Figura 2.4: Exemplo de componentes no espaço YCbCr: (a) componente Y (Luminância); (c) componente Cb (chrominância, diferença do azul); (d) componente Cr (chrominância, diferença do vermelho).

Existem diversos outros espaços de cores, como HSI, CMY(K), CIELAB, etc. [9], usados de acordo com aplicação desejada. Alguns desses espaços separam as componentes da imagem em informações de luminância (intensidade de iluminação) e chrominância (informação de cor). Como o olho humano é mais sensível às variações de intensidade de luz do que às variações de cor [10], muitos métodos de processamento de imagem são aplicados apenas ao canal de luminância.

Apenas para exemplificar: um espaço de cores bastante usado, tanto em processamento de imagens quanto de vídeos, é o $YCbCr$. Este espaço mantém a informação de luminância no canal Y enquanto as informações de cores são armazenadas nas componentes de diferença de cores Cb e Cr . Para se converter uma imagem do espaço RGB para o espaço $YCbCr$, usam-se as seguintes equações, realizando as operações *pixel a pixel*:

$$\begin{aligned}
 Y &= 0.299R + 0.587G + 0.114B \\
 Cb &= 0.564(B - Y) \\
 Cr &= 0.713(R - Y)
 \end{aligned}
 \tag{2.3}$$

2.2.3 Resolução

A resolução de uma imagem representa a quantidade de detalhes discerníveis que ela possui. Existem, contudo, diversas classificações de resolução: resolução de *pixel*, resolução espacial, resolução espectral, resolução temporal e resolução radiométrica [1, 8, 11]. Para o caso de imagens naturais, estas definições estão diretamente relacionadas aos sensores de captura, que convertem energia eletromagnética em *bits*. Trataremos aqui exclusivamente de câmeras que capturam luz no espectro visível.

A resolução de *pixel* é informalmente definida como a quantidade de *pixels* de uma imagem, nas direções horizontal e vertical. Este termo também é usado para definir a quantidade de *pixels* que um dispositivo de visualização (monitor, TV, etc.) pode reproduzir. Ao longo deste trabalho usaremos esta definição para nos referirmos às dimensões espaciais, ou tamanho, de uma imagem. Por exemplo, um quadro de vídeo na resolução de *pixel* (ou dimensões) conhecido por *Full HD* tem 1920×1080 *pixels*.

A resolução espacial refere-se à densidade de *pixels* por unidade de área do sensor de captura no momento da captura da imagem. Esta resolução depende do tamanho do sensor de captura e da quantidade de seus elementos. Assim, quanto maior a densidade de elementos, maior a resolução possível do sistema de imageamento. A resolução determina o menor nível possível de detalhe espacial discernível, pois os detalhes também podem ser limitados por componentes ópticos, como borramentos causados pela lente. Em suma, qualitativamente, a resolução espacial é a menor quantidade de detalhes discerníveis em uma imagem, ao passo que, quantitativamente, pode ser medida de diferentes maneiras, como *pixels* por unidade de distância ou pares de linhas por unidade de distância. Desta forma, uma definição comumente utilizada para resolução espacial é o maior número discernível de pares de linhas por unidade de distância. É importante ressaltar que medidas de resolução espacial, para terem significado, devem ser estabelecidas com respeito a unidades de distância [1, 8].

A resolução temporal é definida como a quantidade de imagens capturadas por instante de tempo. Quando aplicada a vídeo, esta resolução define a taxa de quadros capturados por instante de tempo e, costumeiramente, é medida em quadros por segundo. Para o caso de imagens por satélite, a resolução temporal está relacionada ao período que o satélite leva para imagear uma mesma região observada por um mesmo ângulo.

A resolução espectral determina a habilidade de um sensor em detectar intervalos de comprimento de ondas. Quanto mais estreito esse intervalo, maior a resolução. Câmeras convencionais, por exemplo, têm resolução espectral relativamente baixa, uma vez que capturam apenas os comprimentos de onda referentes às cores vermelha, verde e azul. Alguns sistemas de sensoriamento remoto, por outro lado, possuem sensores chamados hiperespectrais, capazes de capturar distintamente centenas de bandas dentro do espectro eletromagnético visível.

Por fim, a resolução radiométrica, ou resolução de intensidade, é determinada pela sen-

sibilidade de um sensor de captura à magnitude da energia eletromagnética por ele captada. Assim, ela pode ser definida como a menor variação de nível de intensidade que pode ser capturada por um sensor. Esta resolução define o número de níveis de quantização em que o alcance dinâmico de um *pixel* pode ser dividido.

2.3 REDIMENSIONAMENTO DE IMAGEM

A modificação do tamanho de uma imagem se dá pela inserção ou remoção de *pixels*. O processo que aumenta o seu tamanho pela inserção de novos *pixels* é chamado de interpolação. O processo oposto, que diminui o tamanho da imagem pela remoção de *pixels*, é chamado de decimação¹.

2.3.1 Interpolação

O processo de interpolação se dá pela criação de uma nova imagem de maior tamanho a partir dos *pixels* de uma imagem de menor tamanho. A interpolação por um fator de escala $E \in \mathbb{N}$ consiste em inserir entre cada uma das linhas e colunas em uma imagem $f(x, y)$ $E - 1$ linhas e $E - 1$ colunas compostas por zero, respectivamente. De forma mais geral, tomando uma imagem $f(x, y)$, em que x e y são as coordenadas dos *pixels* da imagem nas direções vertical e horizontal, uma imagem $f_I(x, y)$ interpolada a partir de $f(x, y)$ por um fator de escala E é dada por

$$f_I(x, y) = \begin{cases} f\left(\frac{x}{E}, \frac{y}{E}\right) & , x = kE \text{ e } y = lE, k, l \in \mathbb{N} \\ 0 & , \text{ caso contrário} \end{cases} \quad (2.4)$$

Os *pixels* com valor zero podem ser substituídos por valores a partir de *pixels* de valores conhecidos. A técnica mais simples, conhecida por vizinho mais próximo (ou *nearest-neighbor* em inglês), ocorre pela substituição direta de cada zero pelo seu vizinho mais próximo conhecido. Contudo, é possível se obter uma imagem interpolada com melhor qualidade visual ao se substituir os valores zeros inseridos por combinações de valores pré-existentes. Isso é feito pela aplicação de filtros lineares $H(x, y)$.

Diversos filtros podem ser usados na filtragem posterior ao processo de interpolação. Alguns dos filtros mais comuns podem ser definidos por núcleos unidimensionais e implementados de forma separável. Assim, para filtrar uma imagem, aplica-se o filtro primeiramente na direção vertical e depois na direção horizontal. Dentre os mais comuns, temos os filtros bilinear [12], bicúbico [13] e Lanczos [14]. O filtro bilinear é descrito como:

¹Alguns autores usam o termo “dizimação”

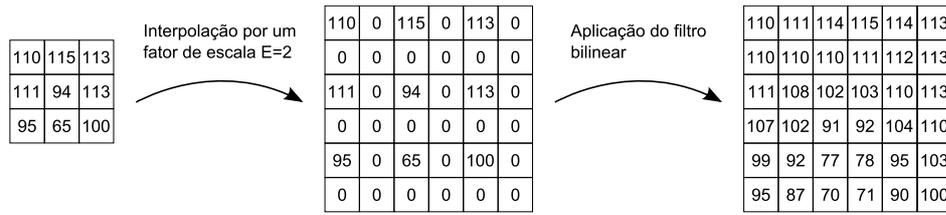


Figura 2.5: Exemplo de um bloco de *pixels* interpolado por um fator de escala $E = 2$ e filtrado com filtro bilinear.

$$h(\alpha) = \begin{cases} 1 - |\alpha|/E, & |\alpha| \leq E, \\ 0, & |\alpha| > 1, \end{cases} \quad (2.5)$$

Já o filtro bicúbico foi originalmente proposto na forma

$$h(\alpha) = \begin{cases} (u+2)|\alpha|^3 - (u+2)|\alpha|^2 + 1, & 0 < |\alpha| < 1 \\ |\alpha|^3 - 5u|\alpha|^2 + 8u|\alpha| - 4u, & 1 < |\alpha| < 2 \\ 0, & |\alpha| > 2 \end{cases} \quad (2.6)$$

A escolha do valor $u = -0,5$ faz com que este filtro possa ser utilizado na aproximação de terceira ordem para a interpolação da imagem original [13]. Por último, o filtro Lanczos é descrito pela equação

$$h(\alpha) = \begin{cases} \text{sinc}(\alpha) \text{sinc}(\alpha/u), & |\alpha| < u \\ 0, & |\alpha| > u \end{cases} \quad (2.7)$$

$$\text{sinc}(\alpha) = \frac{\sin(\pi\alpha)}{\pi\alpha}$$

O parâmetro u é um valor inteiro que determina o tamanho de $h()$, com valores tipicamente de 2 ou 3. Este filtro é uma implementação prática janelada da função $\text{sinc}()$ de interpolação ideal.

A Figura 2.5 mostra um exemplo de um bloco de *pixels* de tamanho 3×3 interpolado por um fator de escala $E = 2$, ou seja, dobrando seu tamanho nas duas dimensões, com a inserção de linhas e colunas de valor zero. É mostrado também o bloco interpolado após a filtragem usando um filtro do tipo bilinear.

A Figura 2.6 mostra um exemplo (já com *zoom*) da interpolação da imagem decimada mostrada na Figura 2.8c com a aplicação dos filtros bilinear, bicúbico (com $u = -0,5$) e Lanczos (com $u = 3$), em comparação com a imagem original.

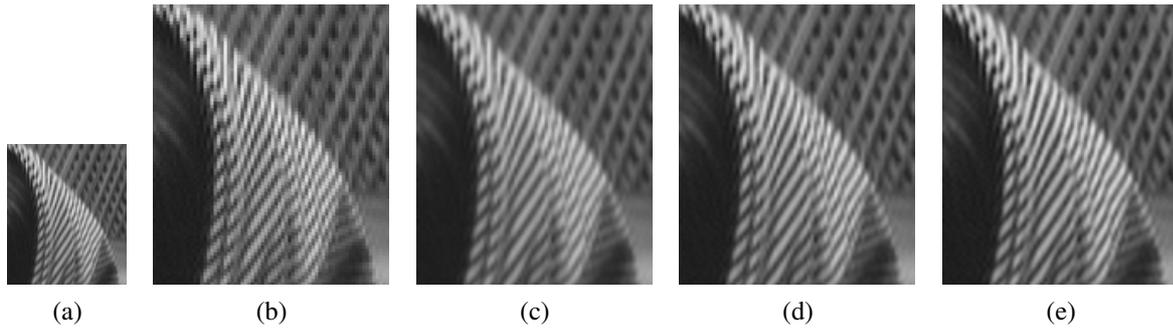


Figura 2.6: Exemplo de interpolação com *zoom*: (a) imagem original; (b) interpolação com vizinho mais próximo; (c) interpolação com filtro bilinear; (d) interpolação com filtro bicúbico; (e) interpolação com filtro lanczos3.

2.3.2 Decimação

O processo de decimação de uma imagem nas direções vertical e horizontal por um fator de escala $E \in \mathbb{N}$ se dá pela manutenção de uma a cada E linhas e uma a cada E colunas, respectivamente, eliminando as demais. De forma mais geral, a versão decimada $f_D(x, y)$ da imagem $f(x, y)$ por um fator de escala E é dada por

$$f_D(x, y) = f(xE, yE). \quad (2.8)$$

Este processo, contudo, pode incorrer em um problema conhecido como superposição espectral (*aliasing*, em inglês) caso a largura de banda da transformada discreta de Fourier da imagem $f(x, y)$ esteja fora do intervalo $\left[-\frac{\pi}{E}, \frac{\pi}{E}\right]$, ou seja, caso a amostragem a uma taxa reduzida não respeite as limitações impostas pelo teorema de Nyquist [15]. Para evitar este efeito, pode-se aplicar um filtro do tipo passa-baixas (neste caso também chamado de *anti-aliasing*) H_{PB} que remova as componentes espectrais fora do intervalo $\left[-\frac{\pi}{E}, \frac{\pi}{E}\right]$ previamente ao processo de decimação. Os filtros usados na decimação podem ser os mesmos usados na interpolação e é comum que sejam referidos como núcleo de borramento (ou *blurring kernel* em inglês). A decimação seguida de filtragem é conhecida como subamostragem [8], pois a imagem com *pixels* removidos $f_D(x, y)$ filtrada é uma representação de uma mesma cena que a imagem $f(x, y)$, porém amostrada a uma taxa de amostragem reduzida, já com o cuidado para remoção de efeito de *aliasing*.

A Figura 2.7 mostra um exemplo de uma imagem que foi decimada por um fator $E = 2$, tanto sem a pré-filtragem quanto com a aplicação do filtro do tipo bicúbico (definido pela equação (2.6), com *zoom* mostrado na Figura 2.8). Note que a imagem decimada sem a filtragem gerou na região do tecido listrado uma textura completamente errônea, ao passo que na textura vista na imagem pré-filtrada este erro é fortemente mitigado.

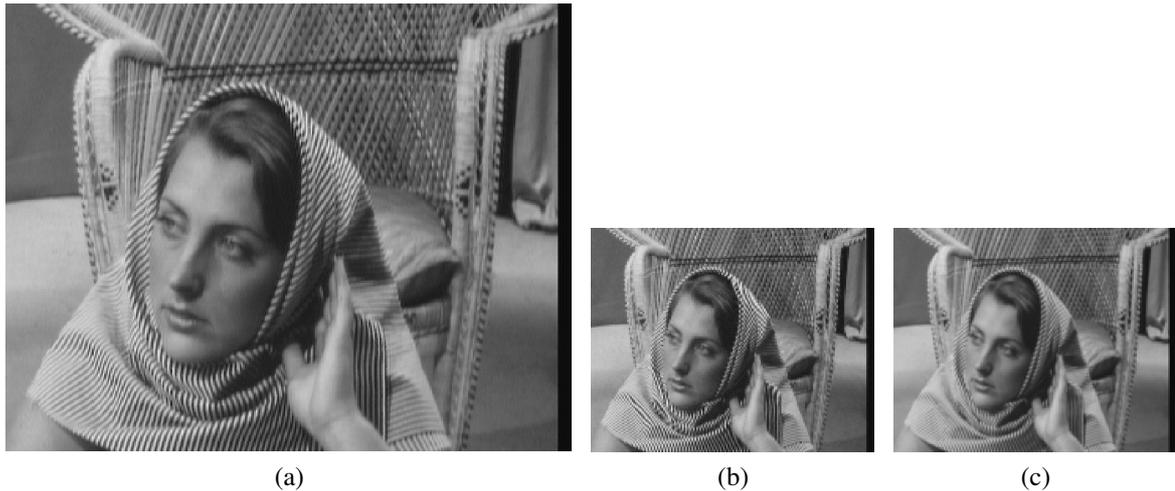


Figura 2.7: Exemplo de decimação: (a) imagem original; (b) imagem decimada sem filtro *anti-aliasing*; (c) imagem decimada com filtro *anti-aliasing* bicúbico.

2.3.3 Relação entre interpolação e decimação

Assim como o processo de decimação seguido de filtragem é conhecido por subamostragem, a interpolação (também já com a filtragem) é conhecida como sobreamostragem. Alguns autores tratam essa terminologia de forma diferente, definido todo o processo de pré-filtragem seguida da eliminação de amostras como decimação. Da mesma forma, definem o processo de inclusão de amostras seguido de filtragem como interpolação [12, 15].

O processo de subamostragem, ao eliminar *pixels* de forma irreversível, diminui a quantidade de detalhes discerníveis da imagem. Isso faz com que a resolução espacial da imagem também seja reduzida. A quantidade de detalhes perdidos depende do filtro usado, ou seja, quanto mais o filtro borra a imagem, maior a redução da resolução espacial. O processo de sobreamostragem, por outro lado, mesmo aumentando o tamanho da imagem, não é capaz de inserir novos detalhes. Por este motivo, a sobreamostragem não é capaz de aumentar a resolução espacial de uma imagem.

Um processo de subamostragem seguido de sobreamostragem (ambos pelo mesmo fator de escala) de uma imagem reduz sua resolução espacial sem, no entanto, reduzir seu tamanho. Usaremos o termo geral “reamostragem” para nos referirmos a esse processo, conforme mostrado na Figura 2.9. Na figura, os filtros $H_D(x, y)$ e $H_I(x, y)$ representam as etapas de filtragem prévia à decimação e posterior à interpolação, respectivamente

A Figura 2.10 mostra um exemplo do mesmo bloco mostrado nas Figuras 2.6 e 2.8, comparando o bloco original com versões reamostradas, com diferentes combinações de filtros de sub e sobre amostragem.

Quando se tem acesso à cena contínua original de que uma imagem digital foi capturada, uma forma de se obter uma imagem de maior tamanho e maior resolução espacial é por superamostragem [8]. Este processo é simplesmente uma nova captura da cena contínua

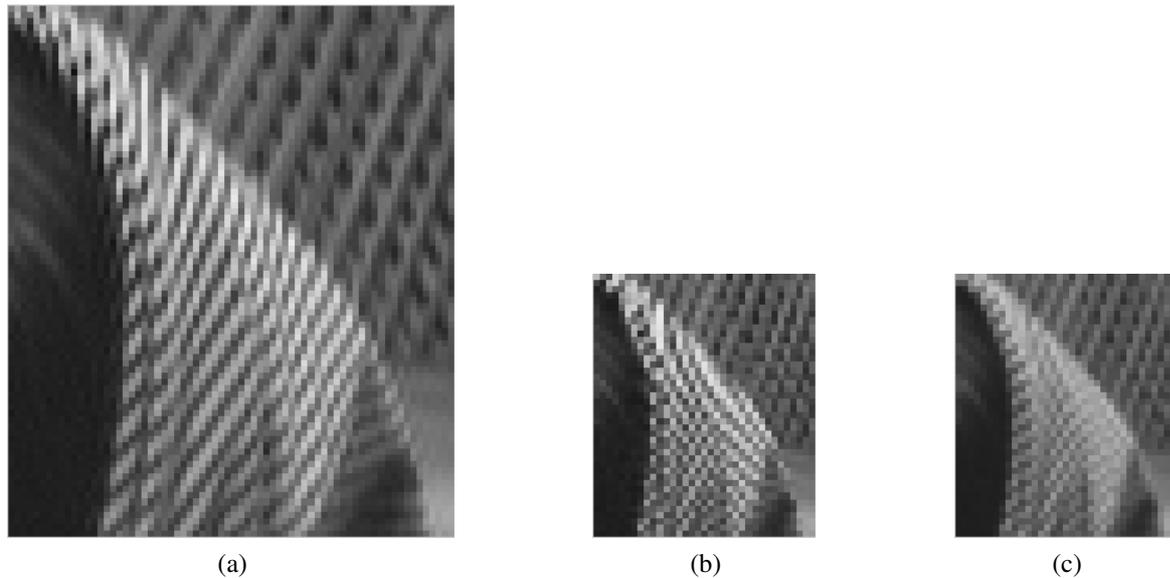


Figura 2.8: Exemplo de decimação com *zoom*: (a) imagem original; (b) imagem decimada sem filtro *anti-aliasing*; (c) imagem decimada com filtro *anti-aliasing* bicúbico.

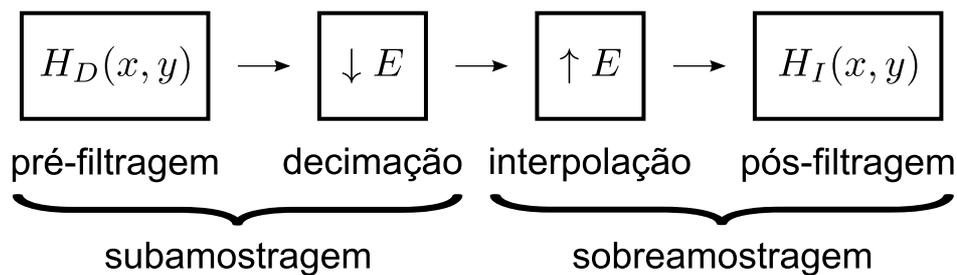


Figura 2.9: Reamostragem como sequência dos processos de subamostragem e sobreamostragem.

com maior taxa de amostragem, ou seja, com um sensor dotado de maior densidade de *pixels*. Como, na maioria das aplicações práticas, temos posse apenas de uma imagem digital (sem qualquer acesso à cena original), podemos buscar outras formas de recuperar os detalhes que tenham sido perdidos, seja no processo de captura, seja por um processo de subamostragem. A este processo de obtenção de uma imagem em alta resolução (AR) a partir de outras imagens dá-se o nome de super-resolução.

2.4 SUPER-RESOLUÇÃO

Conforme foi apresentado no Capítulo 1, o processo de super-resolução (SR) parte inicialmente de uma ou mais imagens em baixa resolução (BR). Para uma melhor compreensão de todo o processo de SR, devemos inicialmente apresentar um modelo de imageamento que relaciona uma imagem original em AR com as imagens observadas em BR.

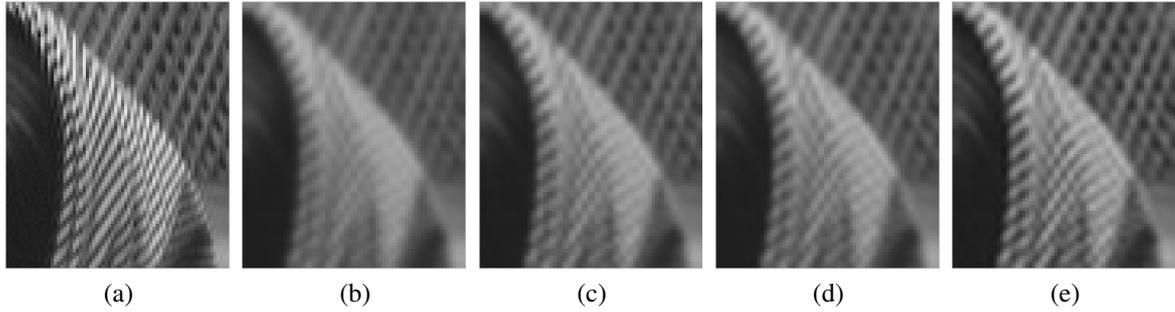


Figura 2.10: Exemplo de reamostragem com diferentes combinações de filtros: (a) imagem original; (b) filtro bilinear para subamostragem e sobreamostragem; (c) filtro lanczos3 para subamostragem e bilinear para sobreamostragem; (d) filtro bilinear para subamostragem e lanczos3 para sobreamostragem; (e) filtro lanczos3 para subamostragem e sobreamostragem.

2.4.1 Modelo de imageamento

Seja a imagem \mathbf{A} em AR de tamanho $L_1N_1 \times L_2N_2$ (ou seja, contendo $L_1N_1 \times L_2N_2$ pixels), escrita como um vetor em forma lexicográfica \mathbf{a} , a imagem que desejamos obter. Seja agora uma imagem \mathbf{B}_k em BR de tamanho $N_1 \times N_2$, escrita como um vetor em forma lexicográfica \mathbf{b}_k , a k -ésima imagem observada de um total de K imagens, com $k \in \{1, 2, \dots, K\}$. Os parâmetros L_1 e L_2 representam as relações de alteração de tamanho entre as imagens \mathbf{A} e \mathbf{B} nas direções vertical e horizontal, respectivamente. A imagem \mathbf{A} é tomada como sendo idealmente não degradada, amostrada acima da taxa de Nyquist e capturada de uma cena contínua assumida como limitada em banda, ou seja, \mathbf{A} não apresenta *aliasing*. Já a imagem \mathbf{B} é resultado de deformação óptica, borramento e decimação aplicados sobre a imagem \mathbf{A} . Assumindo também que cada imagem em BR é corrompida por ruído aditivo, ela pode ser representada pelo modelo de observação [4].

$$\mathbf{b}_k = \mathbf{D}\mathbf{G}_k\mathbf{M}_k\mathbf{a} + \mathbf{r}_k, \text{ para } k = 1, \dots, K. \quad (2.9)$$

Na equação (2.9), \mathbf{M}_k é uma matriz de deformação óptica de tamanho $L_1N_1L_2N_2 \times L_1N_1L_2N_2$. Ela representa o movimento que pode ocorrer durante as aquisições das imagens e pode conter translações, rotações e *zoom*. Esses movimentos podem ser tanto globais (cena inteira) quanto locais (alguns objetos ou regiões específicas da cena). A matriz \mathbf{G}_k tem tamanho $L_1N_1L_2N_2 \times L_1N_1L_2N_2$ e representa o borramento. Este borramento pode ser causado por diversos motivos, como o foco da lente, movimento entre a cena e o sistema de captura e a função de dispersão de ponto, ou PSF (do inglês *point spread function*) do sensor. Já a matriz de decimação \mathbf{D} , de tamanho $(N_1N_2)^2 \times L_1N_1L_2N_2$, representa a diminuição do tamanho e inserção do efeito de *aliasing*. Apesar de o borramento já funcionar de forma similar a um filtro *anti-aliasing*, isso não pode ser garantido em todas as situações. Logo, o seu efeito deve ser levado em consideração. Por fim, \mathbf{r}_k é um vetor de ruído ordenado lexicograficamente.

Este modelo pode ser simplificado pela combinação das três matrizes de deformação óptica (\mathbf{M}_k), borramento (\mathbf{G}_k) e decimação (\mathbf{D}) numa única matriz \mathbf{W}_k de tamanho $(N_1 N_2)^2 \times L_1 N_1 L_2 N_2$ e descrito como:

$$\mathbf{b}_k = \mathbf{W}_k \mathbf{a} + \mathbf{r}_k, \text{ para } k = 1, \dots, K. \quad (2.10)$$

O problema de SR pode então ser resumido como a busca pela imagem \mathbf{a} a partir das imagens \mathbf{b}_k , para $k = 1, \dots, K$. Em geral, para sistemas de imageamento reais, essas matrizes são todas desconhecidas e precisam ser estimadas.

Podemos também representar essas equações 2.10 na forma de um sistema linear

$$\begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \cdot \\ \cdot \\ \mathbf{b}_K \end{bmatrix} = \begin{bmatrix} \mathbf{D}\mathbf{G}_1\mathbf{M}_1 \\ \mathbf{D}\mathbf{G}_2\mathbf{M}_2 \\ \cdot \\ \cdot \\ \mathbf{D}\mathbf{G}_K\mathbf{M}_K \end{bmatrix} \mathbf{a} + \underline{\mathbf{r}}, \quad (2.11)$$

ou, equivalentemente,

$$\underline{\mathbf{b}} = \underline{\mathbf{W}} \mathbf{a} + \underline{\mathbf{r}}, \quad (2.12)$$

em que $\underline{\mathbf{b}}$, $\underline{\mathbf{r}}$ e $\underline{\mathbf{W}}$ representam as concatenações dos vetores \mathbf{b}_k , \mathbf{r}_k e da matriz \mathbf{W}_k , respectivamente, para $k = 1, 2, \dots, K$.

Existem diversas abordagens para se resolver este problema, tais como [1]: interpolação-restauração [16]; abordagem estocástica [17]; Projeção em Conjuntos Convexos [18]; SR no domínio da frequência [2]; e SR baseada em exemplos [5]. Apresentamos aqui, de forma breve, cada uma dessas abordagens. Para uma revisão mais completa, recomendamos que o leitor se refira a [3].

2.4.2 Super-resolução por interpolação-restauração

Esta é a abordagem mais simples para a solução do problema e se baseia em três estágios: registro de imagens; interpolação não-uniforme; e restauração e remoção de ruídos. Supõe-se inicialmente que as imagens \mathbf{B}_k possuam um deslocamento relativo de *subpixels*, ou seja, os *pixels* de uma imagem contêm informação que estaria entre os *pixels* de outra imagem. O registro das imagens permite que elas sejam alinhadas e reposicionadas de acordo com uma grade com o posicionamento dos *pixels* de \mathbf{A} . Como a posição dos *pixels* das imagens \mathbf{B}_k não se encaixa na grade, é necessária uma interpolação não-uniforme para se determinar os valores dos *pixels* nas posições corretas. Por fim, usa-se algum algoritmo de restauração e

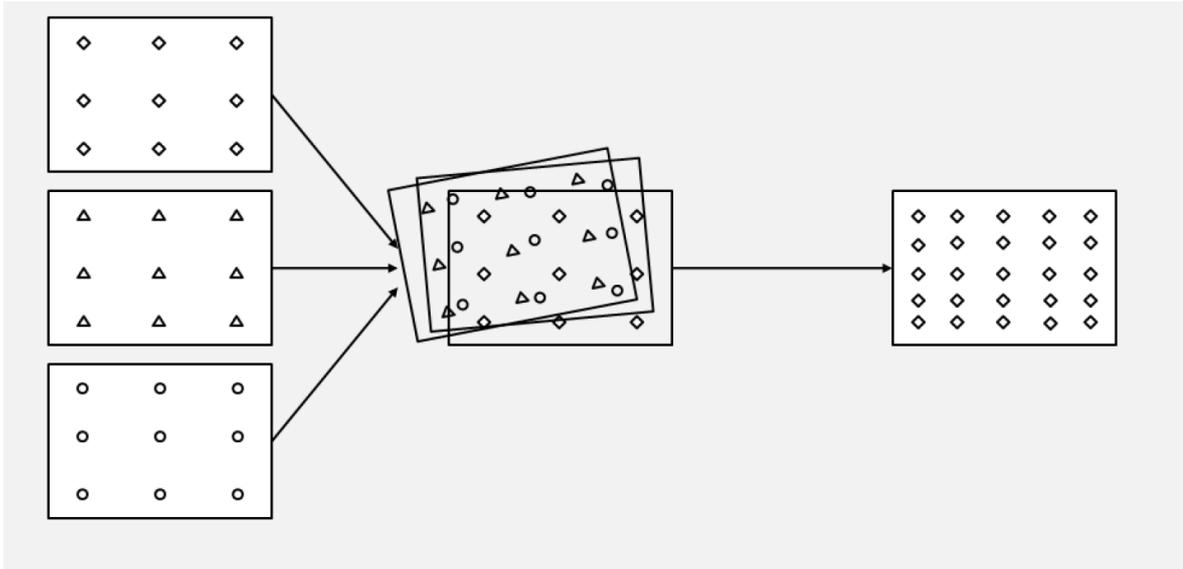


Figura 2.11: SR por interpolação baseada em alinhamento e “desborramento” (adaptada de [1]).

remoção de ruídos. Essas três etapas podem ser realizadas de forma separada ou em conjunto, dependendo das técnicas utilizadas. A Figura 2.11 mostra um exemplo deste processo.

2.4.3 Super-resolução por abordagem estocástica

As soluções usando a abordagem anterior são simples, diretas e intuitivas, quando se assumem modelos de observação simples. Contudo, não se pode garantir a otimalidade das soluções. Com isso, surgiram soluções que tomam tanto a imagem em AR, quanto as deformações ópticas como variáveis estocásticas. Tomando uma matriz de degradação $\mathbf{W}(\nu, h)$ (que agrupa as deformações ópticas, decimações e borramentos referentes a cada imagem observada), com vetor de movimento ν e núcleo de borrimento h , a SR pode ser resolvida por estimação *Bayesiana*:

$$\begin{aligned}
 \mathbf{a} &= \arg \max_{\mathbf{a}} Pr(\mathbf{a}|\mathbf{b}) \\
 &= \arg \max_{\mathbf{a}} \int_{\nu, h} Pr(\mathbf{a}, \mathbf{W}(\nu, h)|\mathbf{b}) d\nu dh \\
 &= \arg \max_{\mathbf{a}} \int_{\nu, h} \frac{Pr(\mathbf{b}|\mathbf{a}, \mathbf{W}(\nu, h)) Pr(\mathbf{a}|\mathbf{W}(\nu, h))}{Pr(\mathbf{b})} d\nu dh \\
 &= \arg \max_{\mathbf{a}} \int_{\nu, h} Pr(\mathbf{b}|\mathbf{a}, \mathbf{W}(\nu, h)) Pr(\mathbf{a}) Pr(\mathbf{W}(\nu, h)) d\nu dh
 \end{aligned} \tag{2.13}$$

em que $Pr(\mathbf{b}|\mathbf{a}, \mathbf{W}(\nu, h))$ é a verossimilhança dos dados, $Pr(\mathbf{a})$ é informação *a priori* da imagem em alta-resolução desejada e $Pr(\mathbf{W}(\nu, h))$ é informação *a priori* da estimação do movimento. Note que \mathbf{a} e \mathbf{W} são estatisticamente independentes.

Esta solução é relativamente complexa. Porém, podem-se assumir algumas condições para resolver este problema [1]. Primeiramente, pode-se assumir que o ruído aditivo seja um

vetor aleatório Gaussiano branco e de média zero, ou seja:

$$Pr(\underline{\mathbf{b}}|\mathbf{a}, \underline{\mathbf{W}}(\nu, h)) \propto \exp\left\{-\frac{1}{2\sigma^2}\|\underline{\mathbf{b}} - \underline{\mathbf{W}}(\nu, h)\mathbf{a}\|^2\right\}. \quad (2.14)$$

Pode-se definir $Pr(\mathbf{a})$ usando uma distribuição de Gibbs na forma exponencial:

$$Pr(\mathbf{a}) = \frac{1}{Z} \exp\{-\rho(\mathbf{a})\}. \quad (2.15)$$

em que $\rho(\mathbf{a})$ é uma função potencial não-negativa, chamada função de energia, e Z é apenas um fator de normalização. Finalmente, assumindo que $\underline{\mathbf{W}}(\nu, h)$ seja previamente estimado (denominado por $\underline{\mathbf{W}}$), a equação (2.13) reduz à formulação da solução do problema de SR por *Maximum a Posteriori* (MAP):

$$\begin{aligned} \mathbf{a} &= \arg \max_{\mathbf{a}} Pr(\underline{\mathbf{b}}|\mathbf{a}, \underline{\mathbf{W}}) Pr(\mathbf{a}) \\ &= \arg \min_{\mathbf{a}} \{\|\underline{\mathbf{b}} - \underline{\mathbf{W}}\mathbf{a}\|^2 + \lambda\rho(\mathbf{a})\} \end{aligned} \quad (2.16)$$

em que $\rho(\mathbf{a})$ impõe um fator de penalização por soluções malformadas [19] e λ pondera essa penalização, enquanto absorve a variância do fator de ruído.

2.4.4 Super-resolução por projeção em conjuntos convexos

O método de projeção em conjuntos convexos, ou POCS (do inglês *projection onto convex sets*), interpreta a solução do problema de SR como membro de um conjunto convexo fechado C_i , que é definido como um conjunto de vetores que satisfaz uma propriedade em particular. Cada informação *a priori* restringe a solução a um conjunto específico. Assim, o problema de SR pode ser formulado pela definição de múltiplos conjuntos convexos restritos que contenham a imagem \mathbf{A} desejada. Diversas restrições podem ser usadas, como restrição de consistência, por exemplo, que assume corretas as informações *a priori* de movimento. Nesta restrição os conjuntos convexos são definidos como:

$$C_k = \left\{ \mathbf{a} \mid \|\mathbf{W}_k \mathbf{a} - \mathbf{b}_k\|^2 \leq \sigma^2, 1 < k < K \right\}, \quad (2.17)$$

nos quais σ reflete o intervalo de confiança de que a imagem realmente pertença ao conjunto C_k e é determinada pela estatística do processo do ruído.

A solução do problema pode ser então encontrada pela intersecção dos conjuntos, caso ela não seja um conjunto vazio, ou seja, $\mathbf{a} \in C_s = \bigcap_{k=1}^K C_k$. Esta solução pode ser encontrada por meio de um algoritmo iterativo

$$\mathbf{a}_{t+1} = P_K P_{K-1} \cdots P_2 P_1 \mathbf{a}_t, \quad (2.18)$$

em que \mathbf{a}_0 é um ponto inicial arbitrário e P_i é um operador de projeção que projeta um ponto em um conjunto convexo fechado C_i .

2.4.5 Super-resolução no domínio da frequência

Esta abordagem relaciona uma imagem em AR com diversas imagens em BR transladadas em uma formulação no domínio da frequência, tirando proveito das propriedades de deslocamento e *aliasing* das transformadas contínua e discreta de fourier, ou CFT (do inglês *continuous fourier transform*) e DFT (do inglês *discrete fourier transform*), respectivamente. Sejam $a(t_1, t_2)$ uma imagem contínua em AR e $a_k(t_1, t_2) = a(t_1 + \delta_{k_1}, t_2 + \delta_{k_2})$ sua k -ésima versão deslocada de valores δ_{k_1} e δ_{k_2} arbitrários, porém conhecidos, com $k = 1, \dots, K$. A CFT da imagem $a(t_1, t_2)$ é dada por $\mathcal{A}(u_1, u_2)$ e as transformadas das imagens deslocadas são dadas por $\mathcal{A}_k(u_1, u_2)$. Pela propriedade de deslocamento da CFT, temos:

$$\mathcal{A}_k(u_1, u_2) = \exp[j2\pi(\delta_{k_1}u_1 + \delta_{k_2}u_2)]\mathcal{A}(u_1, u_2). \quad (2.19)$$

As imagens deslocadas são amostradas com período de amostragem T_1 e T_2 de forma a gerar as imagens observadas em BR $b_k[n_1, n_2] = a_k(n_1T_1 + \delta_{k_1}, n_2T_2 + \delta_{k_2})$ com $n_1 = 0, 1, 2, \dots, N_1 - 1$ e $n_2 = 0, 1, 2, \dots, N_2 - 1$. Tomando as DFTs de cada uma dessas imagens como $\mathcal{B}_k[\Omega_1, \Omega_2]$, a sua respectiva CFT será relacionada pela propriedade de *aliasing*, assumindo que $\mathcal{A}(u_1, u_2)$ seja limitado em banda (ou seja, $|\mathcal{A}(u_1, u_2)| = 0$ para $|u_1| > (N_1\pi/T_1)$ e $|u_2| > (N_2\pi/T_2)$):

$$\mathcal{B}_k[\Omega_1, \Omega_2] = \frac{1}{T_1T_2} \sum_{m_1=-\infty}^{\infty} \sum_{m_2=-\infty}^{\infty} \mathcal{A}_k\left(\frac{2\pi}{T_1}\left(\frac{\Omega_1}{N_1} - m_1\right), \frac{2\pi}{T_2}\left(\frac{\Omega_2}{N_2} - m_2\right)\right). \quad (2.20)$$

Podemos relacionar os coeficientes da DFT $\mathcal{B}_k[\Omega_1, \Omega_2]$ com as amostras da CFT desconhecida de $a(t_1, t_2)$ em forma matricial, unindo as equações 2.19 e 2.20:

$$\underline{\mathcal{B}} = \underline{\Phi}\underline{\mathcal{A}}, \quad (2.21)$$

onde $\underline{\mathcal{B}}$ é um vetor coluna de tamanho $K \times 1$ cujo k -ésimo elemento é o coeficiente da DFT $\mathcal{B}_k[\Omega_1, \Omega_2]$, $\underline{\mathcal{A}}$ é um vetor coluna de tamanho $N_1N_2 \times 1$ contendo as amostras dos coeficientes da CFT desconhecida de $a(t_1, t_2)$, e $\underline{\Phi}$ é uma matriz de tamanho $K \times N_1N_2$ relacionando $\underline{\mathcal{B}}$ e $\underline{\mathcal{A}}$. Finalmente, a reconstrução da imagem em AR desejada demanda que se determine a matriz $\underline{\Phi}$ para então resolver o problema inverso e encontrar $\underline{\mathcal{A}}$. Em seguida, deve-se aplicar a DFT inversa sobre o vetor encontrado.

A grande vantagem dessa solução é a sua simplicidade teórica. Contudo, ela tem sérias limitações, pois assume um modelo de translação global com parâmetros conhecidos e

sem ruídos. Além disso, assume também que o processo de amostragem é impulsivo sem modelagem de efeito de borramento do sensor. Algumas soluções surgiram buscando uma modelagem mais realista, como: a modelagem do borramento para cada imagem [20]; modelo de translação por blocos [21]; e uso da transformada discreta de cossenos, ou DCT (do inglês *Discrete Cosine Transform*) em substituição à DFT [22]. Porém, essas soluções falham em possibilitar que se trabalhe com imagens de degradações mais complexas bem como o uso de informação *a priori* existente apenas no domínio espacial.

2.4.6 Super-resolução baseada em exemplos

Os métodos apresentados anteriormente requerem a posse de diversas imagens em BR capturadas de uma mesma cena e com algumas informações *a priori* conhecidas. A SR baseada em exemplos [5], contudo, requer apenas uma imagem em BR e, por isso, é também conhecida por SR de imagem única. Este método usa um banco de dados de recortes em AR e seus recortes correspondentes em BR para adicionar informação de alta frequência a uma versão interpolada da imagem em BR que se deseja super-resolver.

O banco de dados é composto a partir de dois conjuntos de recortes, $\{\mathbf{A}_k\}_{k=1}^K$ retirados de imagens em AR, e $\{\mathbf{B}_k\}_{k=1}^K$ retirados de imagens em BR correspondentes. Esses recortes podem ter sido extraídos de uma ou diversas imagens. Os pares de recortes $(\mathbf{A}_k, \mathbf{B}_k)$ estão relacionados por um modelo de observação $\mathbf{B}_k = i(d(\mathbf{A}_k + \mathbf{R}))$, em que as funções $d(\cdot)$ e $i(\cdot)$ representam os processos de subamostragem (pré-filtragem seguida de decimação) e sobreamostragem (interpolação seguida de pós-filtragem), respectivamente, e \mathbf{R} é algum ruído. Os filtros de cada um dos processos $d(\cdot)$ e $i(\cdot)$ não precisam ser os mesmos, onde o pré-filtro em $d(\cdot)$ modela a degradação que se deseja desfazer na SR.

Seja \mathbf{X}_I uma versão já sobre-amostrada da imagem em BR que se deseja super-resolver. Seja \mathbf{Y} uma imagem contendo apenas informação de alta frequência, tal que a imagem super-resolvida é dada por $\mathbf{X}_{SR} = \mathbf{X}_I + \mathbf{Y}$. A SR baseada em exemplos é fundamentada na semelhança entre recortes da imagem \mathbf{X}_I e os recortes em BR dos pares $(\mathbf{A}_k, \mathbf{B}_k)$. Tomemos então o j -ésimo recorte \mathbf{X}_I^j de \mathbf{X}_I . Podemos buscar dentre os vetores do conjunto $\{\mathbf{B}_k\}$ aquele $\mathbf{B}_{\hat{k}}$ que mais se assemelhe a \mathbf{X}_I^j , usando alguma métrica de distância D , ou seja, buscamos o índice \hat{k} tal que

$$\hat{k} = \underset{k}{\operatorname{argmin}} D(\mathbf{X}_I^j, \mathbf{B}_k) \quad (2.22)$$

Podemos extrair a informação de alta frequência referente ao recorte encontrado pela diferença entre $\mathbf{B}_{\hat{k}}$ e sua versão em AR $\mathbf{A}_{\hat{k}}$. Para fins práticos, não é necessário armazenar os recortes em AR, mas apenas a diferença entre eles e os em BR, ou seja, recortes contendo apenas informação de alta frequência $\mathbf{C}_k = \mathbf{A}_k - \mathbf{B}_k$. Assim, a informação de alta frequência é calculada para o melhor casamento de recortes como $\mathbf{C}_{\hat{k}} = \mathbf{A}_{\hat{k}} - \mathbf{B}_{\hat{k}}$. Para obter a versão

super-resolvida \mathbf{X}_{SR}^j do recorte \mathbf{X}_I^j , basta somar a informação de alta frequência calculada, ou seja:

$$\mathbf{X}_{SR}^j = \mathbf{X}_I^j + \mathbf{C}_{\hat{k}}. \quad (2.23)$$

Fazendo esse mesmo procedimento para todos os recortes de \mathbf{X}_I , formamos a imagem \mathbf{Y} com todos os recortes $\mathbf{C}_{\hat{k}}$ encontrados e obtemos sua imagem final super-resolvida \mathbf{X}_{SR} . Este procedimento, apesar de simples, traz a ideia geral da SR baseada em exemplos.

A proposta inicial de Freeman *et al.* [5], além dos filtros das etapas de decimação e interpolação, também aplica um filtro passa-altas para remover a intensidade média das imagens (*offset* de baixa frequência), bem como uma normalização de contraste. Isso permite que o mesmo banco de dados de recortes possa ser usado para imagens com diferentes contrastes e intensidades médias. A Figura 2.12 mostra um exemplo de SR baseada em exemplos, contendo a imagem original em BR e suas versões interpolada e super-resolvida. São mostradas também as imagens de onde foram extraídos recortes em AR e os recortes super-resolvidos.

2.4.7 Super-resolução de vídeo

Uma das aplicações de SR é a melhoria da resolução de quadros de vídeo. Este problema costuma ser abordado de duas formas: SR de múltiplas imagens ou SR de única imagem. A SR de múltiplas imagens procura compor quadros em AR a partir dos quadros em BR, comumente usando a abordagem estocástica, como em [23]. Já a SR de uma única imagem busca resolver o problema pela SR baseada em exemplos, ou seja, usando um dicionário com pares de recortes em BR e AR [24, 25].

Neste contexto, introduz-se o conceito de vídeo de resolução mista, o qual é composto por uma sequência de quadros em BR (chamados quadros-não-chave) intercalados por quadros em AR (chamados quadros-chave). Este vídeo pode ser gerado pela redução da resolução de alguns dos quadros de um vídeo que, originalmente, possuía todos os quadros com a mesma resolução. Caso a resolução espacial de um quadro seja suficientemente reduzida usando filtros de borramento, este quadro pode ser decimado sem perda de informação ou inserção de *aliasing* [26]. Assim, esses vídeos podem ser de resolução e tamanho mistos. Uma outra forma de gerar esse tipo de vídeo é pela captura simultânea dos quadros de vídeo em uma dada resolução e fotografias em resolução mais elevada². A Figura 2.13 mostra um exemplo de vídeo de resolução mista, já com quadros-não-chave em tamanho reduzido.

²Este tipo de captura é possível em câmeras comerciais, como câmeras GoPro [27], por exemplo.



Figura 2.12: Exemplo de SR baseada em exemplos (retirado de [5]): (a) imagem em BR; (b) imagem interpolada; (c) imagem super-resolvida; (d) imagem original em AR; (e-g) imagens de referência em AR; (h) recortes da imagem super-resolvida; (i) recortes das imagens de referência usados para obter os recortes super-resolvidos em (h)

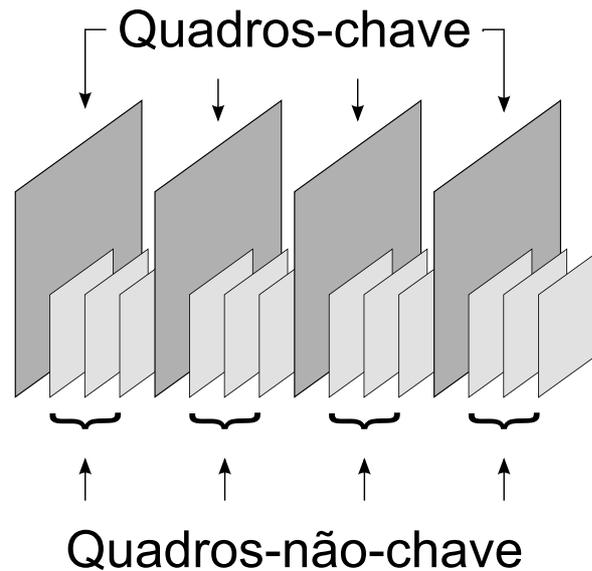


Figura 2.13: Exemplo de uma sequência de vídeo em resolução mista com quadro-chave na resolução original e quadros-não-chave com resolução e tamanho reduzidos.

A existência de quadros/imagens em resoluções diferentes capturados de uma mesma cena permite a abordagem de SR baseada em exemplos para elevar a resolução dos quadros não-chave. A super-resolução possibilita ao vídeo inteiro ser reproduzido em resolução elevada. A redução da resolução de quadros selecionados de um vídeo permite que certos processamentos, como compressão [28], sejam menos dispendiosos computacionalmente. Neste caso, a computação é reduzida porque o processamento é aplicado a imagens menores. Neste tipo de compressão, após o processamento, faz-se necessária a SR para elevar a resolução do quadro à mais próxima de sua resolução original.

Para a super-resolução de quadros-não-chave usando quadros-chave como referência, alguns trabalhos [24, 25] recorrem ao uso de compensação de movimento com blocos sobrepostos, ou OBMC (do inglês *overlapped block motion compensation*) [29]. A estimação de movimento [10] pode ser usada para buscar, em versões em BR dos quadros-chave (decimado-interpolado), por blocos que se assemelhem aos blocos do quadro-não-chave. Os blocos encontrados formam os pares de recorte da SR baseada em exemplo. Quando combinados, os blocos formam imagens compensadas de alta e baixa resoluções. A técnica OBMC é usada para a formação das imagens compensadas, mas se valendo da sobreposição de blocos para mitigar efeito de bloco.

2.4.7.1 Super-resolução de vídeo de resolução mista por ponderação de dicionário

Além do uso de OBMC para o casamento de blocos a serem super-resolvidos, Hung *et al.* [25] apresentaram uma técnica de composição e ponderação de dicionário para a definição do recorte de alta frequência a ser adicionado a cada recorte do quadro-não-chave. Damos um destaque especial para esta técnica porque ela é usada como uma ferramenta em nosso

trabalho.

Seja novamente um bloco \mathbf{X}_I^j do quadro-não-chave \mathbf{X}_I que se deseja super-resolver e o seu bloco colocalizado \mathbf{Y}^j pertencente à imagem de alta frequência \mathbf{Y} . Seja também o dicionário formado pelos pares de blocos em BR \mathbf{A}_k e blocos de alta frequência $\mathbf{C}_k = \mathbf{A}_k - \mathbf{B}_k$, em que \mathbf{B}_k é bloco em AR de que \mathbf{A}_k foi obtido. O dicionário é montado usando estimação de movimento entre o quadro-não-chave \mathbf{X}_I e versões reamostradas dos quadros-chave, seguido da OBMC. Variando parâmetros como o tamanho do bloco na OBMC, são compostas várias imagens compensadas. O dicionário é formado pelos blocos das imagens compensadas colocalizados com o bloco \mathbf{X}_I^j , ou seja, é composto um dicionário para cada bloco de \mathbf{X}_I .

A técnica de ponderação de dicionário busca encontrar pesos $\{\omega_k\}$

$$\operatorname{argmin}_{\{\omega_k\}} D \left(\mathbf{X}_I^j, \sum_{k=1}^K \omega_k \mathbf{B}_k \right) \quad (2.24)$$

tais que

$$\mathbf{Y}^j = \sum_{k=1}^K \omega_k \mathbf{C}_k \quad (2.25)$$

Os autores mostraram que os pesos podem ser calculados por

$$\omega_k = \left(\frac{1}{D_k} \right) \left(\sum_{k=1}^K \frac{1}{D_k} \right)^{-1} \quad (2.26)$$

em que $D_k(\mathbf{X}_I^j, \mathbf{B}_k)$ são as distâncias entre os blocos \mathbf{X}_I^j e \mathbf{B}_k calculadas usando alguma métrica, como a soma de diferenças absolutas, ou SAD (do inglês *sum of absolute differences*) ou a soma de diferenças quadráticas, ou SSD (do inglês *sum of squared differences*), por exemplo. Assim, cada bloco super-resolvido \mathbf{X}_{SR}^j é obtido por

$$\mathbf{X}_{SR}^j = \mathbf{X}_I^j + \mathbf{Y}^j = \mathbf{X}_I^j + \sum_{k=1}^K \omega_k \mathbf{C}_k \quad (2.27)$$

Novamente, a combinação de todos os blocos super-resolvidos leva ao quadro final super-resolvido \mathbf{X}_{SR} .

2.5 AVALIAÇÃO DE QUALIDADE DE IMAGEM E VÍDEO

Técnicas de processamento de imagens e vídeos muitas vezes têm sua avaliação verificada pela qualidade da imagem resultante que produzem. Uma imagem comprimida com

perdas, por exemplo, sofre degradações inerentes ao seu processo e a avaliação da qualidade da imagem é usada para comparar diferentes sistemas de compressão [10].

Medir a qualidade de uma imagem, no entanto, é um processo difícil e normalmente impreciso, sendo essencialmente uma avaliação subjetiva, sujeita a diversos fatores. Dentre esses fatores, estão o ambiente onde a imagem é mostrada, concentração e humor do observador. Buscando normatizar a avaliação subjetiva, a Recomendação BT.500-13 da ITU-R [30] traz diversos procedimentos de testes. Isso leva em consideração situações em que o objetivo final da imagem é ser apresentada a um observador humano, e não a algum processamento de máquina.

É possível, contudo, se fazer uma avaliação objetiva da qualidade de uma imagem ou vídeo que procura aproximar a avaliação subjetiva, de forma a retornar critérios acurados e que possam ser comparados [10]. As métricas de qualidade objetivas se apresentam como uma alternativa à avaliação subjetiva por serem mais simples e bem menos custosas de serem executadas. Uma forma de avaliar a qualidade de uma imagem processada é fazer uma avaliação comparativa da imagem degradada com a sua versão não degradada. Uma imagem super-resolvida, por exemplo, pode ser comparada à sua versão original em AR. Assim, imagens super-resolvidas por diferentes técnicas podem ser comparadas segundo algum critério para avaliar o desempenho de cada técnica.

Uma das métricas mais usadas [10] é a relação sinal-ruído de pico, ou PSNR (do inglês *peak signal-to-noise ratio*). Esta métrica se baseia no erro médio quadrático, ou MSE (do inglês *mean squared error*), entre duas imagens e o compara com o máximo valor de *pixel* possível. Assim, para duas imagens F (com *pixels* $F_{i,j}$) e \hat{F} (com *pixels* $\hat{F}_{i,j}$) ambas de tamanho $M \times N$ e com valores de *pixel* entre 0 e $(2^n - 1)$, em que n é a profundidade de *bits*, o MSE e a PSNR são dados por:

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (F_{i,j} - \hat{F}_{i,j})^2, \quad (2.28)$$

$$PSNR = 10 \log_{10} \left(\frac{(2^n - 1)^2}{MSE} \right). \quad (2.29)$$

Para o caso específico de vídeo, a PSNR é normalmente calculada pela média dos valores calculados para cada quadro. Ainda tratando o caso específico de vídeo, é comum a comparação entre duas ferramentas quaisquer que produzam distorções diferentes para uma mesma taxa de uma forma mais ampla. Em vez de se comparar a diferença entre distorções para uma mesma taxa, traçam-se duas curvas (uma para cada vídeo) e compara-se a diferença entre as curvas [31].

2.6 CARACTERÍSTICAS E DESCRITORES

Em uma imagem, uma característica (ou *feature*, em inglês) é alguma informação de interesse para alguma aplicação específica e está geralmente associada a uma mudança de uma ou mais propriedades da imagem. Uma característica pode ser global, quando diz respeito à imagem inteira, ou local, quando está relacionado à sua vizinhança imediata [32]. Por estar ligada a este trabalho, vamos fazer um apanhado apenas de características locais. Em muitas aplicações, não se tem apenas a necessidade de encontrar (ou extrair) características, mas sim compará-las entre imagens para detecção de objetos, criação de imagens de panoramas, etc. Para isso, podem-se usar descritores, que são formas de representação de características.

Uma das formas de se obter características de imagens é por meio de estruturas que apresentem algum destaque ou algum padrão, como pontos, bordas ou regiões (ou *blob*, em inglês). Bordas (ou contornos) são características já usadas há algum tempo em aplicações como, por exemplo, segmentação. Dentre os detectores de bordas, alguns dos mais usados são os operadores de gradiente e o detector de Canny [33]. Para algumas aplicações, como o registro de imagens, por exemplo, tem-se interesse em detectar cantos de imagens. Alguns dos detectores de cantos mais comuns são as técnicas Harris [34], SUSAN [35] e FAST [36]. Apesar de serem bastante usadas, características como bordas e cantos podem ter suas aplicações limitadas por não serem totalmente invariantes a algumas transformações. Para resolver este problema, foram propostas diversas técnicas para se extrair características invariantes locais de imagens.

2.6.1 Características invariantes locais

Uma característica é dita invariante se ela não se altera sob condição de alguma transformação [32], tal como rotação, translação, escala e afim. Em outras palavras, uma mesma característica deve ser detectada mesmo que a imagem sofra algum tipo de transformação. Características invariantes locais, idealmente, devem pertencer a algum objeto e ter algum significado. Contudo, isso não é sempre factível, pois demanda um alto nível de interpretação de conteúdo da imagem. Além disso, o uso deste tipo de característica está diretamente ligado à correspondência de estruturas entre imagens distintas ou entre versões transformadas de uma mesma imagem. Assim, boas características devem ter algumas das seguintes propriedades:

- Repetibilidade: Para duas imagens de um mesmo objeto ou cena, uma alta porcentagem de características encontradas em uma imagem também deve ser encontrada na outra.
- Distinção: Padrões de intensidade por trás das características detectadas devem ser bem diferentes, de forma que as características possam ser distinguidas e correspondidas.

- **Localidade:** características devem ser locais, de forma a reduzir a probabilidade de oclusão e permitir aproximações por modelos geométricos simplificados.
- **Quantidade:** O número de características detectadas deve ser suficientemente grande de forma que uma quantidade razoável (dependente da aplicação) seja detectada mesmo para objetos pequenos. Além disso, deseja-se a detecção de objetos mesmo sob oclusão.
- **Acurácia:** características detectadas devem ser localizadas, tanto na localização da imagem quanto com respeito à escala.

O processo de detecção dessas características parte da detecção de algum ponto de interesse na imagem. Alguns detectores, como o detector baseado em matriz Hessiana e detector de Harris, já demonstram certa invariância e robustez à rotação, iluminação e ao ruído [32].

O detector baseado em Hessiana (ou simplesmente detector Hessiana), proposto por Beaudet *et al.* [37], usa a segunda derivada da intensidade da imagem para detectar variações de gradiente em direções ortogonais. A matriz Hessiana \mathcal{H} pode ser usada para medir a curvatura em um ponto quando a imagem é tratada como uma superfície de intensidade. Sejam então $f(x, y)$ uma imagem e f_x e f_y suas derivadas parciais da imagem nas direções x e y , respectivamente, conforme apresentado na Subseção 2.2.1. Seja também a imagem suavizada $L(x, y, \sigma) = g(x, y, \sigma) * f(x, y)$, em que $g(x, y, \sigma)$ é um núcleo de suavização Gaussiano de escala σ variável dado por

$$g(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}. \quad (2.30)$$

A matriz Hessiana, acima mencionada, é calculada por:

$$\mathcal{H} = \begin{bmatrix} L_{xx}(x, y, \sigma) & L_{xy}(x, y, \sigma) \\ L_{xy}(x, y, \sigma) & L_{yy}(x, y, \sigma) \end{bmatrix}, \quad (2.31)$$

em que $L_{xx}(x, y, \sigma)$ etc. são as derivadas de segunda ordem da imagem, posteriormente suavizadas pelo núcleo Gaussiano.

O determinante da matriz Hessiana pode ser usado para detectar estruturas de imagens com uma forte variação de sinal em duas direções. As características são finalmente determinadas após um processamento do determinante calculado por supressão de não-máximos, ou seja, é considerado máximo apenas o valor maior que os seus oito vizinhos imediatos.

Proposto por Harris e Stephens [34], o “Detector de Harris”, utiliza a matriz de derivadas da imagem para detectar cantos. A matriz de derivadas é calculada por

$$\mathcal{M} = \begin{bmatrix} L_x^2(x, y, \sigma_D) & L_x(x, y, \sigma_D)L_y(x, y, \sigma_D) \\ L_x(x, y, \sigma_D)L_y(x, y, \sigma_D) & L_y^2(x, y, \sigma_D) \end{bmatrix}, \quad (2.32)$$

Em outras palavras, a matriz de derivadas é composta a partir da suavização das derivadas parciais por um núcleo Gaussiano na escala σ_D , seguido do cálculo dos produtos dessas derivadas suavizadas. Os autovalores da matriz \mathcal{M} indicam as mudanças de sinais principais nas direções ortogonais, formando uma descrição de \mathcal{M} invariante à rotação. Dois autovalores grandes indicam a presença de um canto (mudança nas duas direções), enquanto que apenas um autovalor grande indica uma borda (mudança em apenas uma direção). Como o determinante $\det(\cdot)$ de uma matriz é igual ao produto dos autovalores e o traço $\text{traço}(\cdot)$ é a soma, mede-se uma escore de cantos³ \mathcal{C} em que não há necessidade de calcular os autovalores diretamente:

$$\mathcal{C} = \det(\mathcal{M}) - \lambda \text{traço}(\mathcal{M}), \quad (2.33)$$

com valor típico de $\lambda = 0,04$.

Os detectores de cantos de Harris e por Hessiana, porém, não são invariantes à escala. Neste sentido, foi apresentada a técnica Laplaciano da Gaussiana, ou LoG (do inglês *Laplacian of Gaussian*) [38], para determinar se o ponto de interesse em questão se preserva sob transformações de escala usando o conceito de espaço de escalas [39]. Este conceito se baseia na suavização de uma imagem por núcleos de diferentes tamanhos, gerando imagens menos ou mais borradas. O núcleo Gaussiano da Eq. (2.30) é considerado ótimo [40, 41] para gerar a representação neste espaço.

O LoG de uma imagem é calculado pela expressão:

$$\nabla^2 f(x, y, \sigma) = \sigma^2 (L_{xx}(x, y, \sigma) + L_{yy}(x, y, \sigma)), \quad (2.34)$$

em que L_{xx} e L_{yy} , assim como na matriz Hessiana, são as derivadas de segunda ordem da imagem suavizada $L(x, y, \sigma)$. Uma vez calculado o LoG para cada escala, a etapa seguinte é a detecção de valores extremos (mínimo ou máximo) no espaço 3D (x, y, escala) . Isto é feito comparando o valor de um *pixel* com seus 26 vizinhos, sendo oito vizinhos espaciais na mesma escala e nove em cada uma das escalas vizinhas. A Figura 2.14 mostra o *pixel* (marcado com um X) em $\nabla^2 f(x, y, \sigma)$ sendo comparado com seus 26 vizinhos (marcados com O). Este ponto é considerado um ponto extremo se seu valor absoluto for maior que os valores absolutos de todos os 26 vizinhos.

Na busca por características invariantes a uma maior quantidade de transformações, surgiram técnicas que unem a teoria de espaço de escalas com os Detectores por Hessiana e de Harris, o que deu origem às técnicas conhecidas por Hessiana-Laplaciano e Harris-Laplaciano [42]. Além dessas técnicas, também foram propostos detectores de características baseadas nos Detectores Hessiana e de Harris (Hessiana-Afim e Harris-Afim [43]), mas invariantes à transformação afim. Os autores das quatro técnicas anteriores também pro-

³O termo usado em inglês é *cornerness*.

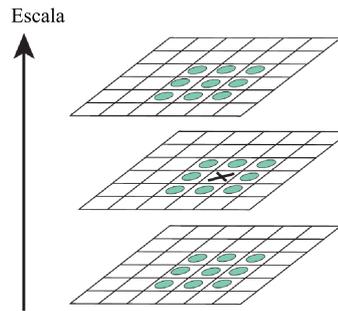


Figura 2.14: Detecção de pontos extremos (adaptado de [26]).

puseram um detector de características invariantes à rotação, escala e transformação afim [44].

Uma vez que uma característica local foi detectada, já garantindo que ela seja invariante às várias transformações, suas informações podem ser codificadas na forma de descritores. Isto permite que se encontrem correspondências (casamentos) entre características de imagens distintas. Alguns dos descritores de características mais antigos estão relacionados a histograma de imagens, como histograma de cor [45], por exemplo. Mais recentemente, foram desenvolvidas novas formas de representação de características por descritores, permitindo que seu casamento seja mais acurado e veloz. Neste contexto, podemos separar os descritores em duas classes: não-binários e binários. Para diversas técnicas, referimo-nos ao tipo de característica e ao seu descritor indiscriminadamente.

2.6.2 Descritores não-binários

Diversos descritores de características têm sido apresentados na literatura, como derivadas de Gaussianas [46], invariantes de momento [47], características complexas [48, 49], *steerable filters* [50] e características locais baseadas em fase [51]. Em um curto espaço de tempo, contudo, foram propostos alguns dos descritores de características mais usados atualmente. O primeiro desses descritores, e talvez o mais conhecido, é a transformada de características invariantes à escala, ou SIFT (do inglês, *Scale Invariant Features Transform*), proposta por Lowe [52, 26], que apresenta desempenho superior aos seus precursores [53].

2.6.2.1 SIFT

Esta técnica detecta, em toda uma imagem, características invariantes à escala e à rotação que também são corretamente casadas para grande variedade de distorções afins, mudanças de ponto de vista 3D, adição de ruído, e mudança de iluminação. SIFT é composta por quatro estágios: detecção de extremos no espaço de escalas; localização do ponto de interesse; atribuição de orientação; e descritor do ponto de interesse. A detecção de extremos no espaço de escalas ocorre por uma implementação baseada na técnica LoG, porém mais eficiente

computacionalmente. Ela é feita pela substituição do operador Laplaciano pelo operador diferença de Gaussianas $D(x, y, \sigma)$, ou DoG (do inglês *difference of Gaussians*), em que k que define a separação entre as escalas:

$$\begin{aligned} D(x, y, \sigma) &= (g(x, y, k\sigma) - g(x, y, \sigma)) * f(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned} \quad (2.35)$$

A partir de $D(x, y, \sigma)$, encontram-se os pontos extremos da mesma forma que na técnica LoG, ou seja, comparando cada *pixel* com seus 26 vizinhos na vizinhança 3D.

A atribuição de orientações é feita a partir do uso de operadores de gradiente locais, o que permite a representação do ponto-chave relativo à sua orientação, tornando-o invariante à rotação. Os gradientes são calculados de forma a serem invariantes à escala, ou seja, as operações de gradiente são realizadas sobre a imagem suavizada $L(x, y, \sigma)$ na escala σ do ponto de interesse encontrado. Para tal, são usados os operadores S_x e S_y :

$$S_x = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}, \quad S_y = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}.$$

Uma vez calculados os gradientes, são calculados os valores de magnitude e ângulo para cada vetor, conforme apresentado na Subseção 2.2.1. É formado então um histograma das orientações (ângulos) dos gradientes em uma região em torno do ponto de interesse. O histograma é composto por 36 *bins*, cobrindo o alcance total de 360° . As direções dominantes são observadas como picos no histograma de orientações, sendo ao maior deles designada a orientação principal. Qualquer outro pico com contagem maior que 80% da contagem do pico principal é usado para criar um novo ponto de interesse com orientação distinta da principal. Dessa forma, para localizações com múltiplos picos de contagem no histograma, são criados pontos-chave com mesma localização e escala, mas com orientações distintas.

Finalmente, os pontos de interesse podem ser representados como descritores. Uma vez determinadas as orientações dos pontos-chave, os gradientes são também usados na representação dos descritores. A Figura 2.15 exemplifica como isso é feito. Do lado esquerdo da figura, são mostrados todos os gradientes em uma região em torno do ponto-chave. A magnitude de cada gradiente é ponderada por uma janela Gaussiana, indicada pelo círculo. Essa janela tem desvio-padrão σ igual à metade do tamanho da janela (8×8 nesse exemplo) e é usada para evitar variações abruptas no descritor para pequenas variações na posição da janela, bem como para dar um peso menor para gradientes mais afastados do ponto-chave. Esses gradientes são então agrupados em um histograma com oito *bins* (representando todos os sentidos ao longo das direções vertical, horizontal e diagonais) para cada sub-região (de

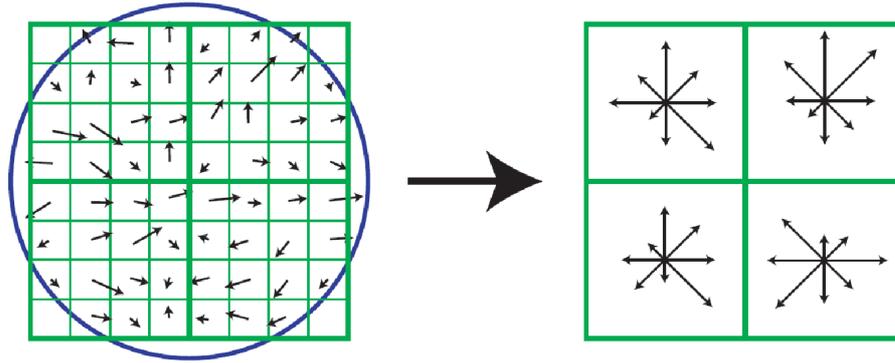


Figura 2.15: Gradientes e descritores (adaptado de [26]).

tamanho 4×4 no exemplo). Os descritores para cada sub-região são então representados como a soma das magnitudes de cada *bin*, conforme representado ao lado direito da Figura 2.15. Os tamanhos da janela e de cada sub-região determinam o tamanho do vetor de característica, ou seja $4 \times 4 \times 8 = 128$. Assim, cada ponto de interesse é representado por um vetor de 128 elementos.

Como cada descritor SIFT contém 128 elementos, Ke *et al.* [54] propuseram uma variação do SIFT usando análise de componentes principais, ou PCA (do inglês, *principal component analysis*). Essa nova técnica reduz a quantidade de elementos para 20.

Outra solução, também baseada em SIFT, é o histograma de localização e orientação de gradiente, ou GLOH (do inglês, *gradient location and orientation histogram*) [53]. Esta técnica propõe computar os descritores SIFT para uma grade log-polar com 17 *bins* e quantizar os gradientes em 16 *bins*, resultando num histograma de 272 *bins*. Usa-se então PCA para reduzir o vetor para 128 elementos.

2.6.2.2 SURF

Propostas por Bay *et al.* [55], as características robustas aceleradas, ou SURF (do inglês, *speeded-Up robust features*), foram apresentadas como uma solução alternativa e com maior eficiência computacional, comparada ao SIFT. Esta técnica busca características baseando-se no detector Hessiana-Laplaciano. Contudo, buscando maior eficiência, ela usa o determinante da matriz Hessiana tanto para detectar a localização espacial quanto a escala. Isto é feito usando um filtro do tipo *box* como aproximação para a derivada de segunda ordem do núcleo Gaussiano, conforme exemplificado na Figura 2.16, que usa um filtro de tamanho 9×9 para aproximar uma Gaussiana com $\sigma = 1,2$. Denotando essas aproximações D_{xx} , D_{yy} e D_{xy} para L_{xx} , L_{yy} e L_{xy} , respectivamente, e a aproximação $\tilde{\mathcal{H}}$ para a matriz Hessiana \mathcal{H} , tem-se o determinante:

$$\det(\tilde{\mathcal{H}}) = D_{xx}D_{yy} - (0,9D_{xy})^2, \quad (2.36)$$

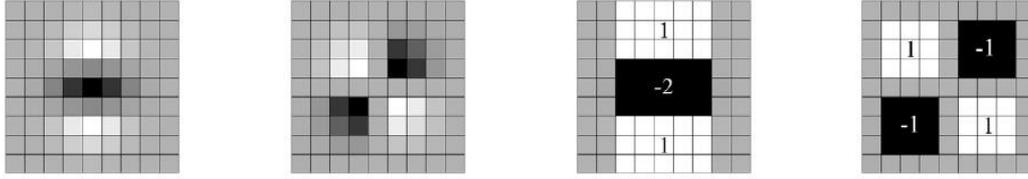


Figura 2.16: Da esquerda para a direita: Derivadas parciais de segunda ordem da Gaussiana nas direções y e xy , e a aproximação das derivadas por filtros do tipo *box* (adaptado de [55]).

Esta aproximação permite que a análise do espaço de escalas seja feita de forma mais eficiente. Em vez de se reduzir a imagem para obtê-la em cada escala iterativamente, pode-se apenas aumentar o tamanho do filtro. Isto permitiria, em princípio, inclusive processamento em paralelo.

O descritor usado no SURF também recorre ao uso de orientação da característica. Isto é feito a partir da resposta da *wavelet* de Haar nas direções x e y da imagem, seguida por uma ponderação por um núcleo Gaussiano centrado no ponto de interesse. A orientação é determinada pela maior soma da resposta ponderada e rotacionada, cobrindo um ângulo de $\frac{\pi}{3}$. O descritor é então definido como a combinação dos vetores $v = (\sum dx, \sum dy, \sum |dx|, \sum |dy|)$ referentes às 16 sub-regiões de tamanho 4×4 pertencentes a um quadrado centrado no ponto de interesse e rotacionado de acordo com a orientação calculada. Os valores de dx e dy correspondem à resposta da *wavelet* de Haar nas direções “vertical” e “horizontal”, referentes à orientação do descritor. Isto resulta num descritor contendo 64 elementos (quatro elementos do vetor v para cada uma das 16 sub-regiões) que representa cada uma das características detectadas.

2.6.3 Descritores binários

Para algumas aplicações, existe a necessidade de que o casamento de descritores ocorra de forma rápida ou que seja limitado o espaço de memória que os armazene. Assim, tanto a quantidade de elementos do descritor, bem como a sua representação numérica eficiente, podem ser consideradas um problema. A técnica de PCA-SIFT [54] foi uma das primeiras a buscar solucionar este problema, por meio da redução de dimensionalidade. Além dela, foi mostrado que os valores dos descritores, quando representados em ponto flutuante, podem ser quantizados sem perda significativa de desempenho [56]. Contudo, uma forma de resolver este problema é gerar descritores binários diretamente de regiões de uma imagem. Essas técnicas são bastante recentes e as principais são BRIEF [57], ORB [58], BRISK [59], FREAK [60] e RIFF [61] e serão brevemente descritas.

Os descritores binários trazem algumas características em comum [62]:

- o descritor é composto pelo conjunto de comparações entre pares de intensidades;
- cada *bit* no descritor é resultado de exatamente uma comparação;

- o padrão de comparação é fixo (exceto por possíveis variações de escala e rotação);
- a medida de similaridade entre valores usada é a distância de Hamming;

Proposto em 2010 por Calonder *et al.* [57], o descritor das características elementares independentes robustas e binárias, ou BRIEF (do inglês *binary robust independent elementary features*) foi o primeiro dos descritores de características invariantes locais e binários propostos. Os autores usam o detector de características usado na técnica SURF, mas pode-se usar qualquer outra característica. Esta técnica é a mais simples e usa padrões de amostragem de 128, 256 ou 512 comparações (resultando em descritores de 128, 256 ou 512 *bits*, respectivamente) com pontos escolhidos aleatoriamente de uma distribuição Gaussiana isotrópica centrada na característica detectada. Este descritor é limitado por não ser invariante à rotação.

O descritor FAST orientado e BRIEF rotacionado, ou ORB (do inglês *oriented FAST and rotated BRIEF*), proposto em 2011 por Rublee *et al.* [58], usa o detector de cantos FAST (junto com a supressão de não-máximos do detector de Harris) e supera a limitação do BRIEF quanto à invariância à rotação. A invariância à rotação é obtida pela designação de uma orientação usando 256 comparações determinadas via aprendizado de máquina que maximiza a variância do descritor enquanto minimiza a correlação sob várias mudanças de orientação.

Já o descritor dos pontos de interesse escalonáveis invariantes robustos e binários, ou BRISK (do inglês *binary robust invariant scalable keypoints*), proposto por Leutenegger *et al.* [59] também em 2011, usa o detector de cantos AGAST [63] (que possui o mesmo desempenho que o detector FAST, porém mais rápido) e é invariante tanto à escala quanto à rotação. A invariância à escala é obtida pela detecção de cantos no espaço de escalas, juntamente à execução de supressão de não-máximos entre as escalas. Já a invariância à rotação é alcançada pela designação de orientação usando padrões simétricos concêntricos em torno da característica detectada. A orientação é determinada a partir das diferenças entre longas distâncias entre pontos (pontos localizados em lados opostos do padrão). O descritor é finalmente definido como 512 comparações em curtas distâncias do padrão (rotacionado e escalonado segundo a orientação definida).

O descritor FREAK (do inglês *fast retina keypoint*), ou ponto de interesse de retina rápido, proposto em 2012 por Alahi *et al.* [60], é inspirado na retina do sistema visual humano. Nesta técnica, as características são detectadas também usando o detector de cantos AGAST. Em seguida, ela computa vetores binários cascadeados comparando intensidades da imagem num padrão semelhante ao das distribuições de células da retina do olho humano. Similarmente, o descritor RIFF (do inglês *retina-inspired invariant fast feature descriptor*), ou descritor de característica rápido e invariante inspirado em retina, proposto por Wu *et al.* [61] em 2014, também usa um padrão de amostragem de pontos baseado na resposta da retina do olho humano, mas com acurácia melhorada em relação ao FREAK. Além disso, o RIFF apresenta considerável ganho de desempenho sobre os demais descritores por sua invariância

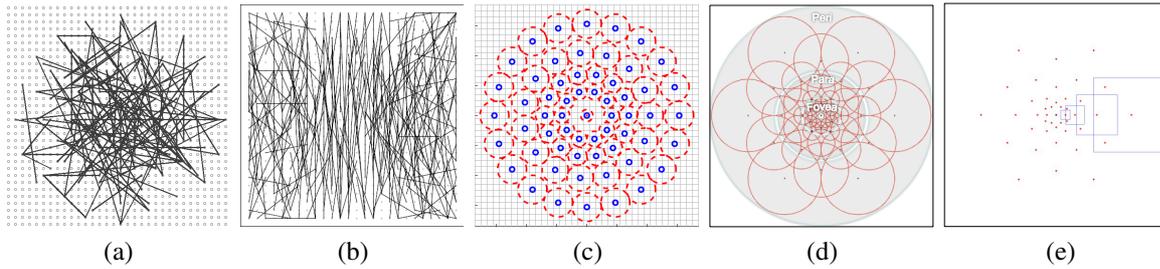


Figura 2.17: Exemplos de padrões de amostragem de pontos para descritores binários: (a) BRIEF; (b) ORB; (c) BRISK; (d) FREAK; (e) RIFF.

à transformação afim. A Figura 2.17 mostra exemplos de padrões de amostragem para cada um dos descritores binários.

Apesar de todos os avanços apresentados por descritores, estudos comparativos mostram que o descritor SIFT ainda supera alguns dos demais, quando de transformações geométricas (translação, rotação, afim, etc.). Quando comparado [64] com os descritores PCA-SIFT e SURF, SIFT supera os demais para transformações de escala, rotação e borrimento, tem desempenho semelhante para transformação afim e é superado para mudança de iluminação. Por outro lado, quando comparado [62] com os descritores SURF, BRIEF, ORB e BRISK, SIFT foi o melhor, exceto para transformações não-geométricas (variação de iluminação, adição de ruído, compressão, etc.). Com relação ao tempo de execução e consumo de memória para armazenamento de descritores, o mais rápido e eficiente é o descritor BRIEF. Nota-se claramente que a escolha do descritor a ser usado depende diretamente da aplicação, do tipo de transformação esperada nas imagens e da necessidade ou não de processamento em tempo real.

2.7 GEOMETRIA PROJETIVA 2D E TRANSFORMAÇÕES BIDIMENSIONAIS

Para duas imagens capturadas de uma mesma cena, a geometria projetiva descreve a forma como essas duas imagens se relacionam de acordo com seus pontos de vista. Usando esta geometria, uma imagem pode sofrer uma transformação bidimensional de forma a posicioná-la do mesmo ponto de vista da segunda imagem. Para isso, é necessário descrever o plano projetivo \mathbb{P}^2 . A geometria projetiva 2D é então definida como o estudo da geometria no plano projetivo \mathbb{P}^2 , podendo também ser definida como o estudo das propriedades do plano projetivo \mathbb{P}^2 que são invariantes sob um grupo de transformações chamadas projetividades [65]. No contexto de super-resolução, geometria projetiva é importante para entender como uma imagem em AR pode ser usada para super-resolver outra em BR capturada de uma mesma cena, porém com diferentes pontos de vista, *zoom* ou disposição de objetos, entre outros.

Uma forma de entender o plano projetivo \mathbb{P}^2 é considerá-lo como um conjunto de raios

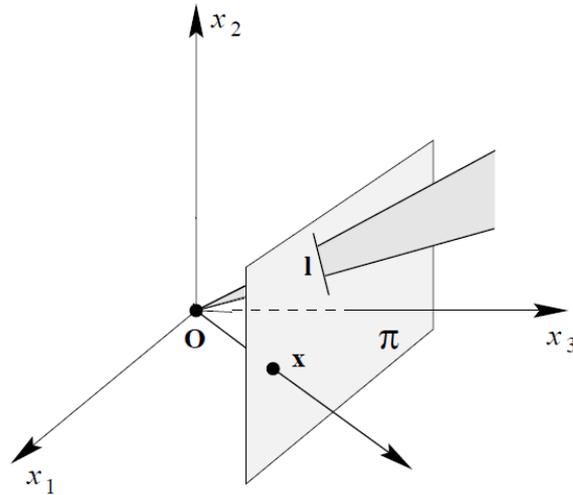


Figura 2.18: Modelo do plano projetivo (adaptado de [65]).

no espaço \mathbb{R}^3 . Sejam então (x_1, x_2, x_3) as coordenadas de um ponto em \mathbb{R}^3 . O conjunto dos vetores $k(x_1, x_2, x_3)$, com k variando, formam um raio (como um raio de luz) partindo da origem (exceto para $k = 0$). Por pertencerem ao mesmo raio, esses vetores são considerados equivalentes e conhecidos como vetores homogêneos. Cada raio pode ser pensado como a representação de um único ponto no plano \mathbb{P}^2 . Neste modelo, as linhas em \mathbb{P}^2 são planos passando pela origem. Pontos e linhas podem ser obtidos pela intersecção desse conjunto de raios e planos passando pelo plano $x_3 = 1$, conforme ilustrado na Figura 2.18. Nesta figura, \mathbf{x} e \mathbf{l} representam um ponto e uma linha no plano π .

De maneira formal, um mapeamento $h : \mathbb{P}^2 \rightarrow \mathbb{P}^2$ é uma projetividade se, e somente se, existe uma matriz não singular 3×3 \mathbf{H} tal que, para qualquer ponto \mathbb{P}^2 representado por um vetor $\mathbf{x} = (x_1, x_2, x_3)$, é verdade que $h(\mathbf{x}) = \mathbf{H}\mathbf{x}$. Projetividade também é conhecida como transformação projetiva, transformação de perspectiva ou homografia. Doravante, usaremos apenas o termo homografia para nos referirmos às projetividades. Uma outra forma de enxergar uma homografia é como uma transformação linear em vetores homogêneos por uma matriz não singular 3×3 :

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \Leftrightarrow \mathbf{x}' = \mathbf{H}\mathbf{x}. \quad (2.37)$$

Podemos agora entender transformações de imagens usando geometria projetiva. Enquanto um ponto no espaço projetivo é representado pelas coordenadas (x_1, x_2, x_3) , um ponto no espaço Euclidiano é representado pelas coordenadas (x, y) . Precisamos relacionar os dois sistemas de coordenadas e uma forma é por meio da equação da reta no espaço Euclidiano, que é definida pela equação $ax + by + c = 0$, em que a escolha dos parâmetros a , b e c define retas distintas. Assim, definimos uma reta pelo vetor $\mathbf{l} = (a, b, c)^T$. Esta

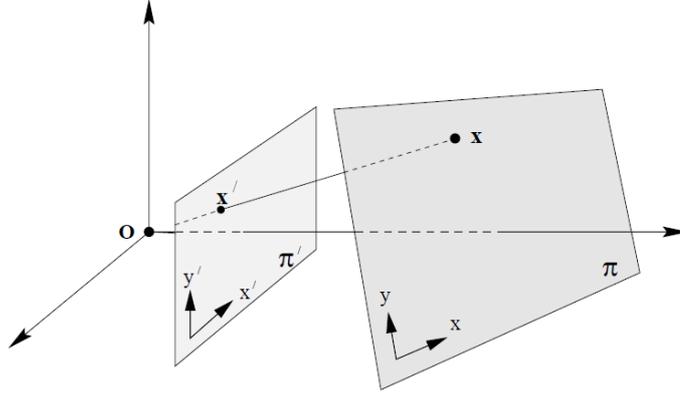


Figura 2.19: Mapeamento entre os pontos de dois planos (retirado de [65]).

equação da reta pode ser reescrita na forma de produto interno, ou seja $(x, y, 1)(a, b, c)^T = (x, y, 1)\mathbf{l} = 0$. Note que para qualquer valor de k não-zero, $(kx, ky, k)\mathbf{l} = 0$ se, e somente se, $(x, y, 1)\mathbf{l} = 0$. Assim, um vetor homogêneo arbitrário na forma $\mathbf{x} = (x_1, x_2, x_3)^T$ representa o ponto $(x_1/x_3, x_2/x_3)^T$ em \mathbb{R}^2 . Sejam então dois sistemas de coordenadas não-homogêneas (x, y) e (x', y') nos planos π e π' , respectivamente. O mapeamento entre os dois sistemas pode ser feito usando a equação (2.37), exemplificado na Figura 2.19, o que nos dá:

$$\begin{aligned} x' &= \frac{x'_1}{x'_3} = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \\ y' &= \frac{x'_2}{x'_3} = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}} \end{aligned} \quad (2.38)$$

2.7.1 Classes de homografia

Se (x, y) e (x', y') representam as posições de *pixels* em duas imagens diferentes, as equações 2.38 representam transformações bidimensionais entre as duas imagens. As transformações projetivas podem ser subdividida em classes, representando cada classe pelas quantidades que são preservadas, ou seja, invariantes. A primeira classe, das isometrias, são transformações do plano \mathbb{R}^2 que preservam as distâncias Euclidianas. Tais transformações são representadas por:

$$\begin{pmatrix} x' \\ y' \\ 1' \end{pmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & t_x \\ \sin \theta & \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \Leftrightarrow \mathbf{x}' = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{x}. \quad (2.39)$$

em que $\mathbf{x} = (x, y, 1)^T$ e $\mathbf{x}' = (x', y', 1)^T$, \mathbf{R} é uma matriz 2×2 de rotação, \mathbf{t} é um vetor 2×1 de translação e $\mathbf{0}$ é um vetor 2×1 nulo. Casos especiais desta são rotação pura (quanto $\mathbf{t} = \mathbf{0}$) ou translação pura (quando $\mathbf{R} = \mathbf{I}$, em que \mathbf{I} é a matriz identidade). Os invariantes

dessa transformação são comprimento (distância entre dois pontos), ângulo entre duas retas e área.

A classe das transformações de similaridade são as isometrias com escalonamento isotrópico (igual em todas as direções), representado por s , ou:

$$\begin{pmatrix} x' \\ y' \\ 1' \end{pmatrix} = \begin{bmatrix} s \cos \theta & -s \sin \theta & t_x \\ s \sin \theta & s \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \Leftrightarrow \mathbf{x}' = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{x}. \quad (2.40)$$

Esta transformação também é conhecida por transformação equi-forma, pois preserva a forma geral da imagem. As invariantes dessa transformação são o ângulo entre retas, o paralelismo entre retas, a razão entre dois comprimentos e a razão entre duas áreas.

Temos, em seguida, a classe das transformações afins, representada por uma transformação linear não-singular seguida por uma translação, ou:

$$\begin{pmatrix} x' \\ y' \\ 1' \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \Leftrightarrow \mathbf{x}' = \begin{bmatrix} \mathbf{\Lambda} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{x}. \quad (2.41)$$

A matriz $\mathbf{\Lambda}$ pode ser decomposta em rotações e escalonamentos não-isotrópicos, ou seja, $\mathbf{\Lambda} = \mathbf{R}(\theta)\mathbf{R}(-\phi)\mathbf{D}\mathbf{R}(\phi)$, em que $\mathbf{R}(\theta)$ e $\mathbf{R}(\phi)$ são rotações pelos ângulos θ e ϕ , respectivamente, e \mathbf{D} é a matriz diagonal:

$$\mathbf{D} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}, \quad (2.42)$$

com λ_1 e λ_2 sendo os fatores de escala nas direções x e y , respectivamente. Os invariantes da transformação afim são o paralelismo entre retas, a razão dos comprimentos de segmentos de retas paralelas e a razão entre áreas.

Por fim, temos a forma mais geral da transformação de perspectiva, cuja única invariante é a razão de razões de comprimentos de retas:

$$\mathbf{x}' = \begin{bmatrix} \mathbf{\Lambda} & \mathbf{t} \\ \mathbf{v}^T & v \end{bmatrix} \mathbf{x}, \quad (2.43)$$

em que $\mathbf{v} = (v_1, v_2)^T$ é responsável pela não linearidade da homografia de forma a permitir a modelagem de pontos de fuga (representação da intersecção de duas retas paralelas), conforme o exemplo da Figura 2.20. Além disso, a matriz costuma ser normalizada para que $v = h_{33} = 1$.

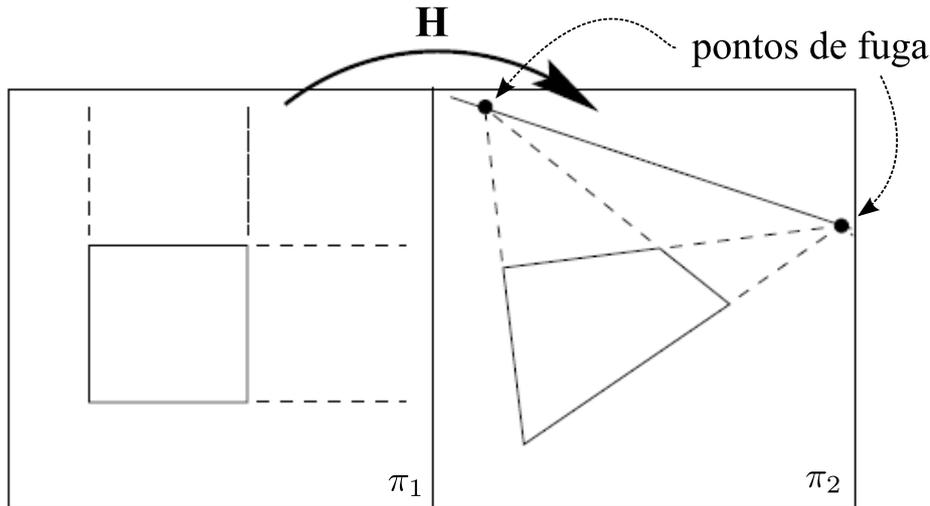


Figura 2.20: Exemplo de mapeamento de um quadrado entre dois sistemas de coordenadas representados pelos planos π_1 e π_2 , em que são mostrados os pontos de fuga (adaptado de [65]).

2.7.2 Definindo uma homografia a partir de pontos de imagens

Como vimos, uma matriz de homografia mapeia os pontos de uma imagem para um outro sistema de coordenadas, compondo uma imagem transformada. Em diversas aplicações, como em construção de imagens panorâmicas, por exemplo, é necessário determinar a matriz a partir de pontos dentre duas imagens. Assumindo que os pontos que pertençam às duas imagens já tenham sido casados (usando correspondência de características, por exemplo), são necessários pelo menos quatro pares de pontos, sendo pelo menos três não-colineares [65]. A Figura 2.21 mostra um exemplo de transformação de perspectiva obtida pela correspondência entre os pontos marcados nas duas imagens.

Quando se tem exatamente quatro pares de pontos, a matriz de homografia pode ser calculada usando o algoritmo da Transformação Linear Direta, ou DLT (do inglês *Direct Linear Transformation*). Quando se tem o casamento de mais de quatro pares de pontos, é necessário usar alguma outra técnica aproximada. Algumas das técnicas mais usadas são as de estimação robusta, como consenso de amostras aleatórias, ou RANSAC (do inglês *random sample consensus*) [66]. Esse tipo de técnica visa estimar uma matriz de homografia a partir de diversos pontos desconsiderando pares de pontos discrepantes (do inglês *outliers*), ou seja, pares de pontos não mapeados pela matriz estimada. Assim, algoritmos como o RANSAC, quando aplicados à estimação de homografia, recebem diversos pares de pontos correspondentes e retornam uma matriz bem como se os pares de pontos são ou não discrepantes.

No próximo capítulo, descrevemos como esses conceitos aqui apresentados são usados para propor uma nova solução de super-resolução baseada em exemplos. A solução proposta usa a correspondência de características para derivar funções de transformação bidimensionais e aplicá-las em uma imagem em AR.



(a)



(b)

Figura 2.21: Exemplo de transformação de perspectiva (retirado de [65]).

Capítulo 3

Formulação do problema e estrutura geral da solução proposta

O problema de super-resolução de imagens pode ser abordado de diversas formas. Conforme descrito por outros autores e mostrado no Capítulo 2, uma das formas de super-resolver uma imagem em baixa resolução é usando informações de contornos presentes em imagens de alta resolução. Neste trabalho, vamos nos concentrar em imagens de baixa resolução que possam ser super-resolvidas a partir de outras que observem uma mesma cena. Contudo, não é nosso foco o trabalho de determinar se as imagens são de fato da mesma cena. Nós partimos do pressuposto de que esta decisão já tenha sido tomada. Um exemplo intuitivo desta premissa é o uso de quadros de um vídeo com imagens sendo tomadas de uma mesma cena, porém em instantes de tempo distintos. Vídeos estereoscópicos ou de múltiplas vistas também se encaixam neste contexto.

A nossa solução surge como uma melhoria àquelas propostas baseadas em estimação e compensação de movimento, mais especificamente às que recorrem ao uso da técnica OBMC, previamente mencionada no Capítulo 2. As técnicas de super-resolução que usam OBMC se baseiam na semelhança entre *pixels*. Num contexto em que a imagem a ser super-resolvida e as imagens de referência são capturadas de uma mesma cena, técnicas baseadas em estimação e compensação de movimento estão fortemente limitadas à forma como cada imagem percebe a cena. Como essas técnicas são executadas por meio de translação de blocos de *pixels*, a super-resolução não obtém resultados tão bons quando as variações entre as imagens são mais complexas, como rotações, *zoom*, transformações afins e de perspectiva.

A nossa solução parte do pressuposto de que dispomos de imagens capturadas de uma mesma cena, mas que podem ter inúmeras variações entre elas. Enquanto as técnicas baseadas em OBMC buscam a semelhança entre *pixels*, a nossa busca a semelhança entre características. Para isso, optamos por usar características SIFT, pois são invariantes à escala, o que é fundamental quando buscamos características que se preservem tanto em uma imagem de alta resolução quanto em uma de baixa. Além disso, conforme mencionado no Capítulo 2, são também invariantes à rotação, são corretamente casadas para grande variedade de distorções afins, mudanças de ponto de vista 3D, adição de ruído e mudança de iluminação, e têm

desempenho superior a outros descritores. As Figuras 3.1a e 3.1b mostram dois exemplos de imagens em baixa e alta resoluções, respectivamente, com descritores SIFT sobrepostos. A Figura 3.2 ilustra a correspondência entre os descritores das duas imagens mostradas nas Figuras 3.1a e 3.1b. Note que a imagem da Figura 3.1a contém mais descritores resultantes de artefatos provocados pela reamostragem da imagem.

Alguns trabalhos trouxeram uma abordagem de super-resolução usando a correspondência de descritores SIFT entre imagens para a obtenção de matrizes de homografia para o cálculo de transformação de imagens (transformação afim ou de perspectiva) para realizar a super-resolução. Cada trabalho usa esse embasamento sob diferentes contextos e para diferentes aplicações de SR. Yuan *et al.* [67] propuseram uma técnica de super-resolução por interpolação-restauração que usa descritores SIFT para derivar a matriz de deformação óptica do modelo de imageamento a partir de várias imagens de baixa resolução de uma mesma cena. Amintoose *et al.* [68] apresentaram uma técnica de super-resolução de imagem única para super-resolver regiões de uma imagem em baixa resolução pela fusão de imagens de alta resolução transformadas. Nemra *et al.* [69] usaram correspondência de descritores SIFT para criar mosaicos de resolução mais elevada que as imagens de entrada no formato interpolação-restauração. Hsu *et al.* [70] também propuseram uma técnica de super-resolução de imagem única pela busca em uma base de dados de recortes usando descritores SIFT. Mais recentemente, inclusive posteriormente ao início do nosso trabalho, Yue *et al.* [71] propuseram uma técnica de super-resolução baseada em exemplos que usa descritores SIFT para realização de busca de imagens em um grande banco de dados (contendo 2496 imagens) semelhantes a uma imagem que se queira super-resolver.

Como buscamos encontrar informações de contornos bem definidos, além da correspondência entre características, recorreremos também ao casamento de gradientes. Assim, o nosso trabalho é distinto das demais propostas de super-resolução baseada em correspondência de descritores SIFT pela forma que buscamos a informação de alta frequência e pela aplicação em que inserimos nossa solução.

- Primeiramente, fazemos uma compensação de movimento usando a correspondência de descritores para derivar matrizes de homografia usando apenas uma ou duas imagens de referência em alta resolução. No caso de aplicação em quadros de vídeo, essas imagens em alta resolução são chamadas de quadros-chave, enquanto que o quadro em baixa resolução a ser super-resolvido é chamado quadro-não-chave.
- Em seguida, buscamos uma melhor aquisição de informação de alta frequência usando casamento de gradientes.
- Por fim, aplicamos nossa solução no contexto de vídeos de resolução mista.

Introduziremos inicialmente algumas notações que usaremos ao longo de todo o trabalho desenvolvido. Seja primeiramente uma imagem original em alta resolução denotada por *Org*. A partir desta imagem, podemos considerar sua versão em baixa resolução como sendo sua versão subamostrada, ou seja, $Org^{sub} = sub(Org)$, usando uma restrição



(a)



(b)

Figura 3.1: Exemplo de descritores SIFT sobrepostos a imagens em (a) baixa resolução e (b) alta resolução.

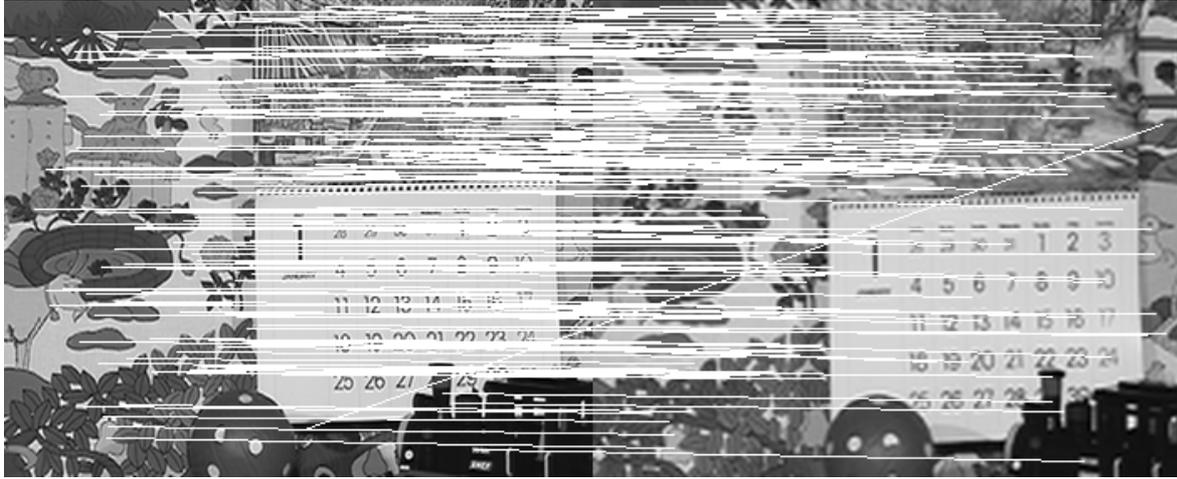


Figura 3.2: Correspondência dos descritores mostrados na figura 3.1.

do modelo de imageamento previamente apresentado no Capítulo 2. Neste caso, a função $sub(\cdot)$ indica o processo de subamostragem, ou seja, decimação precedida por filtragem tipo passa-baixas e desconsideramos a deformação óptica. Assim, supomos que dispomos apenas da imagem Org^{sub} .

O problema que buscamos resolver é super-resolver a imagem em baixa resolução Org^{sub} de forma que se aproxime ao máximo de sua versão em alta resolução Org (da qual não temos posse). Para isso, usamos uma imagem de referência em alta resolução Ref que tenha sido capturada da mesma cena que Org^{sub} e que tenha as mesmas dimensões de Org . Buscamos então alguma imagem $Bord^{SR}$ que contenha apenas informação de alta frequência (bordas ou contornos) extraída de Ref . Assim, a imagem $Bord^{SR}$ será adicionada a uma imagem Org^{baixa} , que é uma versão em baixa resolução, porém de mesmas dimensões que Org . Em outras palavras, Org^{baixa} é uma versão interpolada da imagem Org^{sub} , ou seja, $Org^{baixa} = sobre(Org^{sub}) = sobre(sub(Org))$, em que $sobre(\cdot)$ representa uma função de sobreamostragem (interpolação por zeros seguida de filtragem). Esta soma nos fornece a imagem super-resolvida $Org^{SR} = Org^{baixa} + Bord^{SR}$.

Nas próximas seções, apresentaremos a estrutura geral da nossa solução, bem como alguns detalhes de implementações. Esses detalhes são comuns aos dois diferentes métodos que propomos para nossa solução, os quais serão detalhados nos dois próximos capítulos. Como todas as implementações foram feitas usando o programa MATLAB[®][72] e a biblioteca OpenCV[73], faremos menção a algumas funções específicas utilizadas.

3.1 COMPENSAÇÃO DE IMAGEM BASEADA EM CORRESPONDÊNCIA DE CARACTERÍSTICAS SIFT

A primeira etapa da nossa solução é a compensação de duas imagens, nos moldes da compensação de movimento clássica, mas compreendendo movimentos mais complexos que simples translações. Partimos do pressuposto de que dispomos apenas de duas imagens Org^{baixa} e Ref . A partir dessas duas imagens, compomos uma nova imagem compensada $Comp$ como uma estimativa de Org , porém usando apenas os *pixels* da imagem Ref . Além de $Comp$, compomos também sua versão em baixa resolução $Comp^{baixa}$, que deve ser semelhante a Org^{baixa} . Para isso, geramos a imagem em BR Ref^{baixa} como uma versão reamostrada da imagem Ref , ou seja, $Ref^{baixa} = sobre(sub(Ref))$. Esta etapa é composta dos seguintes passos, ilustrados na Figura 3.3, que serão detalhados em seguida:

- Detecção de características SIFT nas imagens Org^{baixa} e Ref ;
- Correspondência dos descritores das características detectadas;
- Composição de um fluxo de vetores a partir da diferença de posição dos descritores correspondidos;
- Separação do fluxo em grupos de vetores e definição de regiões em torno de cada grupo;
- Definição de homografias a partir dos grupos de vetores;
- Transformação de perspectiva da imagem Ref ;
- Composição de um mosaico pelo agrupamento de recortes das imagens transformadas.

Primeiramente capturamos as características SIFT das imagens Org^{baixa} e Ref , conforme exemplificado nas Figuras 3.1a e 3.1b, respectivamente. Em seguida, fazemos as correspondências dessas características para determinar quais estão presentes em ambas as imagens. Isto é feito usando as técnicas de *best-bin-first* e *nearest-neighbors*, conforme proposto por Lowe [26]. Tanto a detecção de características como a correspondência é feita usando a implementação da biblioteca OpenSIFT, desenvolvida por Hess [74]¹. A correspondência é feita nos dois sentidos, para garantir que exatamente as mesmas características estejam presentes nas duas imagens e foi ilustrada na Figura 3.2.

A diferença entre posições de cada par de características correspondentes nos fornece um vetor de movimento de característica, ou seja, qual o deslocamento sofrido pela característica entre uma imagem e outra. Com isso, compomos um vetor de quatro dimensões na forma $[x, y, v_x, v_y]^T$, que chamamos de vetor de correspondência. Neste vetor, x e y são as coordenadas de uma característica na imagem Org , enquanto v_x e v_y são as componentes do vetor de movimento em direção à característica correspondente na imagem Ref . O conjunto de todos os vetores de correspondência compõe um fluxo de vetores, conforme ilustrado na Figura 3.4.

¹Código disponível em <http://robwhess.github.io/opensift/>

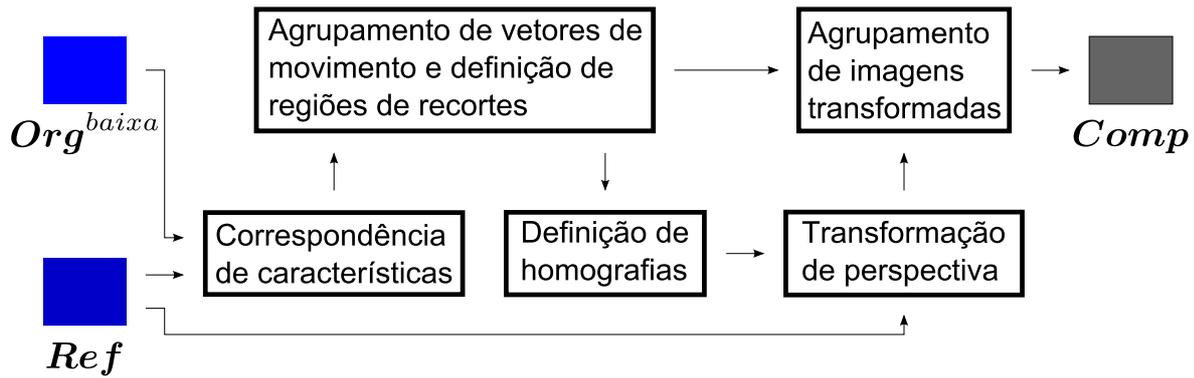


Figura 3.3: Diagrama geral da compensação de movimento baseada em correspondência de características.

Em compensação de movimento clássica, a imagem *Ref* seria simplesmente transladada. Contudo, com um conjunto de vetores, podemos calcular uma matriz de homografia e realizar transformações mais complexas usando transformações de perspectiva. Note que cada vetor $[x, y, v_x, v_y]^T$ pode ser convertido em um par de pontos (x, y) e $(x + v_x, y + v_y)$. Desta forma, por termos possivelmente bem mais que quatro pares de pontos, usamos estimação robusta RANSAC para definir a homografia (transformação) $\tau\{\cdot\}$ que descreve o movimento global médio da cena. Esta estimação é feita usando a função *findHomography* da biblioteca OpenCV. Calculamos então a imagem compensada *Comp* como uma versão transformada da imagem *Ref*, ou seja, $Comp = \tau\{Ref\}$. Usando a mesma função $\tau\{\cdot\}$ derivada do fluxo, calculamos também $Comp^{baixa} = \tau\{Ref^{baixa}\}$.

Apesar de simples, esta compensação pode ser bastante eficiente em situações em que as imagens *Org^{baixa}* e *Ref* apresentem apenas um mesmo objeto sobreposto em um mesmo fundo estático, de forma que o movimento global médio descreve o movimento do objeto. Uma vez que este não é o caso para a maioria das imagens naturais, podemos gerar a imagem compensada *Comp* levando em consideração movimentos distintos presentes na cena.

A imagem compensada *Comp* pode então ser construída como um mosaico de recortes obtidos a partir de versões deformadas da imagem *Ref*. Para isso, é necessário definir as regiões dos recortes que compõem o mosaico, bem como as transformações de perspectiva usadas para deformar a imagem *Ref*. Tanto as regiões de recortes quanto as transformações estão diretamente relacionadas à separação do fluxo de vetores em grupos. A forma como o fluxo é dividido em grupos será descrita nos próximos capítulos. A Figura 3.5 demonstra um exemplo de agrupamento (dos vetores mostrados na imagem 3.4) com 3 grupos de vetores bem como as bordas das suas respectivas regiões de recorte, sobrepostos à imagem *Org^{baixa}*.

Seja então G o número de grupos em que dividimos o fluxo de vetores e g o índice referente a cada grupo, com $g \in \{1, 2, \dots, G\}$. Cada vetor pertence a apenas um grupo, ou seja, não há intersecção entre os grupos. Cada grupo de vetores define como a imagem *Ref* deve ser transformada para que seus *pixels* assumam posições na imagem *Comp*. A partir dos vetores pertencentes a cada grupo g , derivamos uma função $\tau_g\{\cdot\}$. Com isso, geramos cada



Figura 3.4: Exemplo de vetores de correspondência sobrepostos à imagem Org^{baixa} .

versão transformada $\tau_g\{\mathbf{Ref}\}$ da imagem \mathbf{Ref} usando a função *warpPerspective*, também da biblioteca OpenCV. Esta função trata, por interpolação, de questões como mapeamento de diferentes *pixels* de \mathbf{Ref} para uma mesma coordenada em $\tau_g\{\mathbf{Ref}\}$, bem como coordenadas em $\tau_g\{\mathbf{Ref}\}$ às quais não são atribuídos valores.

Em seguida, também a partir dos grupos de vetores, definimos as regiões de recorte que formam o mosaico. Para cada recorte, é definida uma máscara binária M_g , em que a região referente ao grupo de vetores g assume valor 1 enquanto que as demais assumem valor 0. Finalmente, a imagem \mathbf{Comp} pode ser composta pelo agrupamento de versões transformadas da imagem \mathbf{Ref} recortadas, ou seja,

$$\mathbf{Comp} = \sum_{g=1}^G M_g \circ \tau_g\{\mathbf{Ref}\}, \quad (3.1)$$

em que \circ simboliza um produto matricial elemento-por-elemento (ou produto de Hadamard). A Figura 3.6 ilustra um exemplo de como a imagem \mathbf{Comp} é gerada a partir da separação do fluxo nos grupos mostrados na Figura 3.5, com $G = 3$. A Figura 3.7 mostra a imagem \mathbf{Comp} resultante. Usando as mesmas máscaras M_g e funções de transformação $\tau_g\{\cdot\}$, compomos também a imagem compensada em baixa resolução \mathbf{Comp}^{baixa} a partir da imagem \mathbf{Ref}^{baixa} :

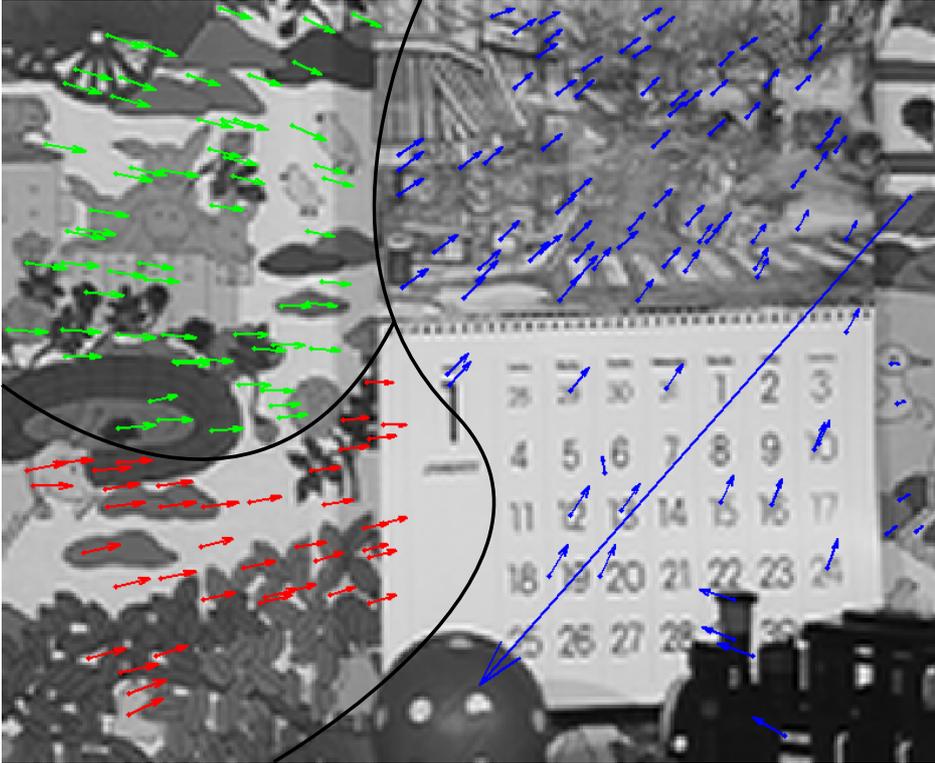


Figura 3.5: Exemplo de agrupamento do fluxo de vetores em três grupos e bordas das regiões de recorte.

$$\mathbf{Comp}^{baixa} = \sum_{g=1}^G \mathbf{M}_g \circ \tau_g \{ \mathbf{Ref}^{baixa} \}, \quad (3.2)$$

As duas ações, definir os grupos de vetores e definir as regiões de recortes, podem ser executadas em ordens distintas: pode-se definir uma região da imagem e determinar os vetores internos a ela; ou podem-se definir os grupos de vetores seguido das regiões que os englobam. Cada um dos métodos apresentados nos capítulos seguintes usa uma ordem distinta de precedência dessas ações.

Uma vez calculado o fluxo de vetores de casamento, podemos definir diferentes formas de agrupar os vetores. Para cada forma de agrupamento distinto, incluindo o fluxo todo como um único grupo (movimento global médio), é criado um par de imagens $\mathbf{Comp}(k)$ e $\mathbf{Comp}^{baixa}(k)$ distinto dos demais, em que k é usado para indexar cada par.

Com o agrupamento das diferentes imagens compensadas, geramos dois conjuntos $\{ \mathbf{Comp}(k) \}$ e $\{ \mathbf{Comp}^{baixa}(k) \}$.

Para aplicações de super-resolução, podemos usar mais de uma imagem de referência \mathbf{Ref}_n , com $n \in \{1, 2, \dots, N-1, N\}$, em que N é o número de imagens de referência usadas. Neste caso, o processo de compensação resulta em duas famílias de conjuntos indexados $\{ \mathbf{Comp}(k) \}_n$ e $\{ \mathbf{Comp}^{baixa}(k) \}_n$.

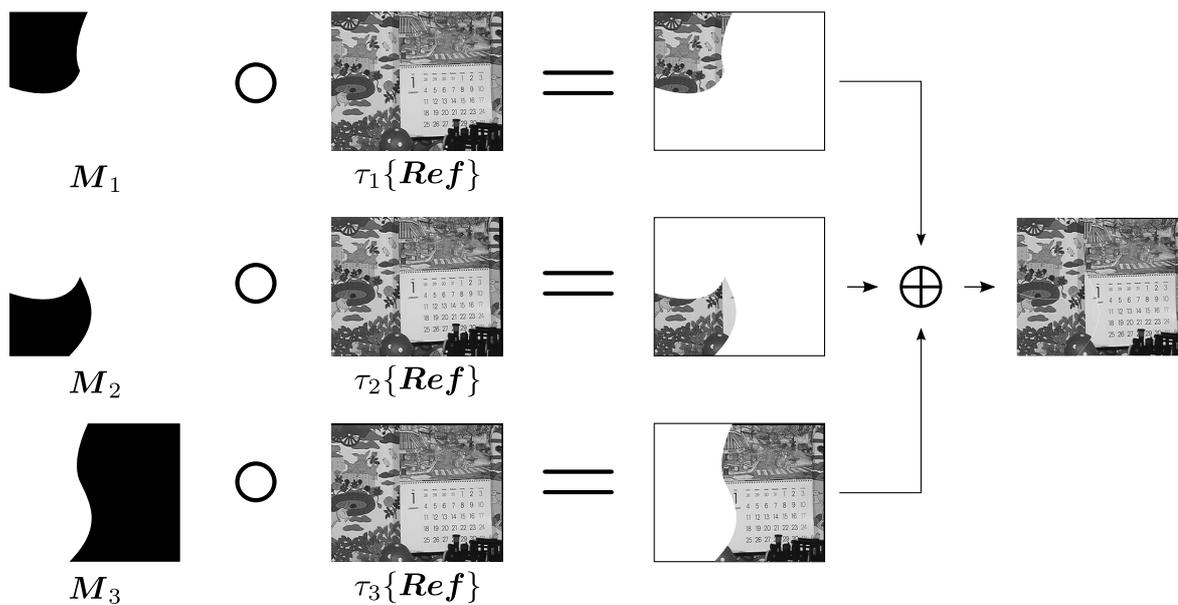


Figura 3.6: Exemplo de composição da imagem $Comp$ a partir das máscaras binárias e versões distorcidas da imagem Ref .



Figura 3.7: Exemplo de imagem $Comp$ resultante da compensação de movimento.

3.2 CONSTRUÇÃO DE IMAGEM POR CASAMENTO DE GRADIENTES

Apesar de as imagens compensadas $\{Comp^{baixa}(k)\}$ na etapa anterior serem semelhantes à imagem Org^{baixa} , podemos melhorar nossa busca por informação de alta frequência para a super-resolução. Para isso, recorreremos ao casamento de gradientes entre as imagens no conjunto $\{Comp^{baixa}(k)\}$ e a própria imagem Org^{baixa} para construirmos, primeiramente, uma imagem aperfeiçoada em baixa resolução $Aper^{baixa}$. Para uma dada posição (i, j) de um *pixel*, tomamos $Org_{i,j}^{baixa}$ como um *pixel* em Org^{baixa} , $Comp_{i,j}^{baixa}(k)$ como um *pixel* em $Comp^{baixa}(k)$ e $\nabla Org_{i,j}^{baixa}$ e $\nabla Comp_{i,j}^{baixa}(k)$ como seus gradientes, respectivamente. Buscamos pelo índice de melhor casamento $\hat{k}_{i,j}$ que satisfaça

$$\hat{k}_{i,j} = \underset{k}{\operatorname{argmin}} \|\nabla Org_{i,j}^{baixa} - \nabla Comp_{i,j}^{baixa}(k)\|. \quad (3.3)$$

Ao encontrarmos esses casamentos, para todas as posições de *pixels*, compomos a imagem aperfeiçoada $Aper^{baixa}$, em que cada *pixel* é dado por $Aper_{i,j}^{baixa} = Comp_{i,j}^{baixa}(\hat{k}_{i,j})$. A partir deste casamento, construímos também uma versão em alta resolução $Aper$ da imagem $Aper^{baixa}$, usando os *pixels* das imagens do conjunto $\{Comp(k)\}$. Em outras palavras, cada *pixel* $Aper_{i,j}$ da imagem $Aper$ é dado por $Aper_{i,j} = Comp_{i,j}(\hat{k}_{i,j})$.

Contudo, uma forma de se obter um casamento mais consistente é considerar uma vizinhança V em torno de cada *pixel*. Isso permite que sejam observados mais gradientes de uma mesma borda e desconsiderados casamentos inverossímeis. Isso é obtido buscando o índice que indica o melhor casamento como

$$\hat{k}_{i,j} = \underset{k}{\operatorname{argmin}} \sum_{s \in V} \sum_{t \in V} \|\nabla Org_{i+s,j+t}^{baixa} - \nabla Comp_{i+s,j+t}^{baixa}(k)\|. \quad (3.4)$$

Como a imagem aperfeiçoada $Aper^{baixa}$ resultante depende da vizinhança, podemos usar um índice v para cada uma, o que nos fornece distintas imagens $Aper^{baixa}(v)$. Quando agrupadas, essas imagens compõem um conjunto $\{Aper^{baixa}(v)\}$. A Figura 3.9 ilustra um exemplo de casamento de gradiente do *pixel* mostrado como um ponto e considerando a vizinhança demarcada em torno dele. Com isso, compomos também o conjunto $\{Aper(v)\}$.

No caso do uso de mais de uma imagem de referência Ref_n , o casamento pode ser realizado usando cada um dos conjuntos indexados $\{Comp^{baixa}(k)\}_n$, o que dá origem a outra família de conjunto indexados $\{Aper^{baixa}(v)\}_n$. Uma outra opção é usar a união dos conjuntos $\{Comp(k)\}_n$ no casamento. Assim, assumindo N imagens de referência, podemos gerar um novo conjunto de imagens compensadas, de forma que

$$\{Comp(k)\}_{N+1} = \bigcup_{n=1}^N \{Comp(k)\}_n. \quad (3.5)$$

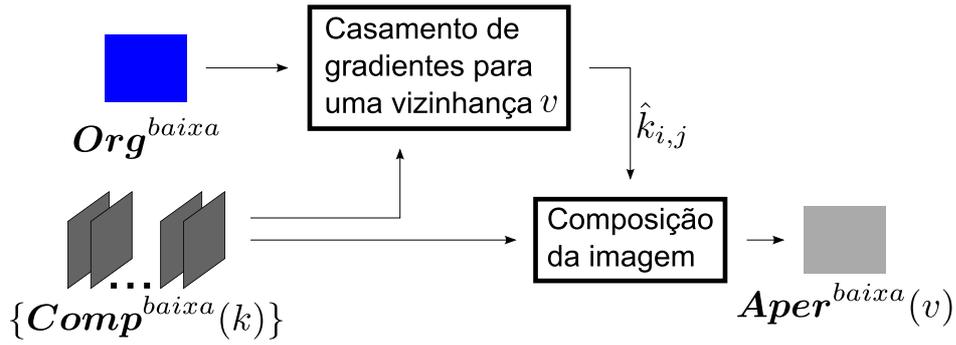


Figura 3.8: Diagrama de operações para a composição de imagem por casamento de gradientes.

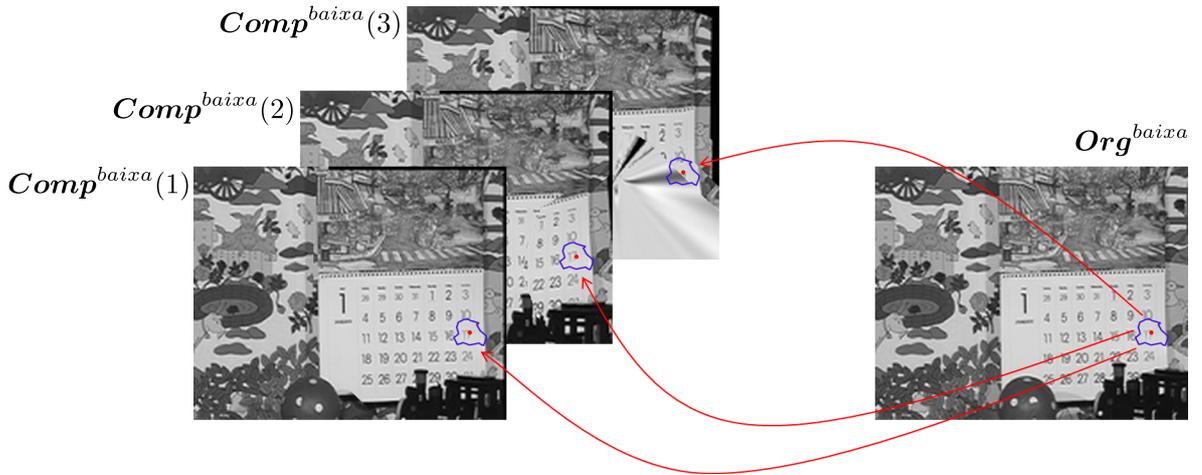


Figura 3.9: Exemplo de casamento de gradientes, considerando uma vizinhança do *pixel*.

Usando o conjunto $\{Comp(k)\}_{N+1}$ no casamento de gradientes, produzimos um outro conjunto $\{Aper(v)\}_{N+1}$ a ser considerado membro da família indexada de conjuntos de imagens aperfeiçoadas.

3.3 COMBINANDO COMPENSAÇÃO POR CORRESPONDÊNCIA DE CARACTERÍSTICAS E CASAMENTO DE GRADIENTES PARA SUPER-RESOLUÇÃO

Apresentamos agora como as duas técnicas descritas anteriormente podem ser combinadas para super-resolução de uma imagem. Na primeira etapa, da compensação de imagens, partimos das imagens *Ref* e Org^{baixa} para produzirmos os conjuntos $\{Comp(k)\}$ e $\{Comp^{baixa}(k)\}$. Na segunda etapa, do casamento de gradientes, usamos a imagem Org^{baixa} juntamente com os conjuntos $\{Comp(k)\}$ e $\{Comp^{baixa}(k)\}$ para compormos os conjuntos $\{Aper(v)\}$ e $\{Aper^{baixa}(v)\}$.

Feito isso, em seguida calcularemos as imagens contendo informação de alta frequência (bordas e contornos) $Bord(v) = Aper(v) - Aper^{baixa}(v)$. Para N imagens de referência, temos os conjuntos $\{Bord(v)\}_n = \{Aper(v)\}_n - \{Aper^{baixa}(v)\}_n$, em que a subtração

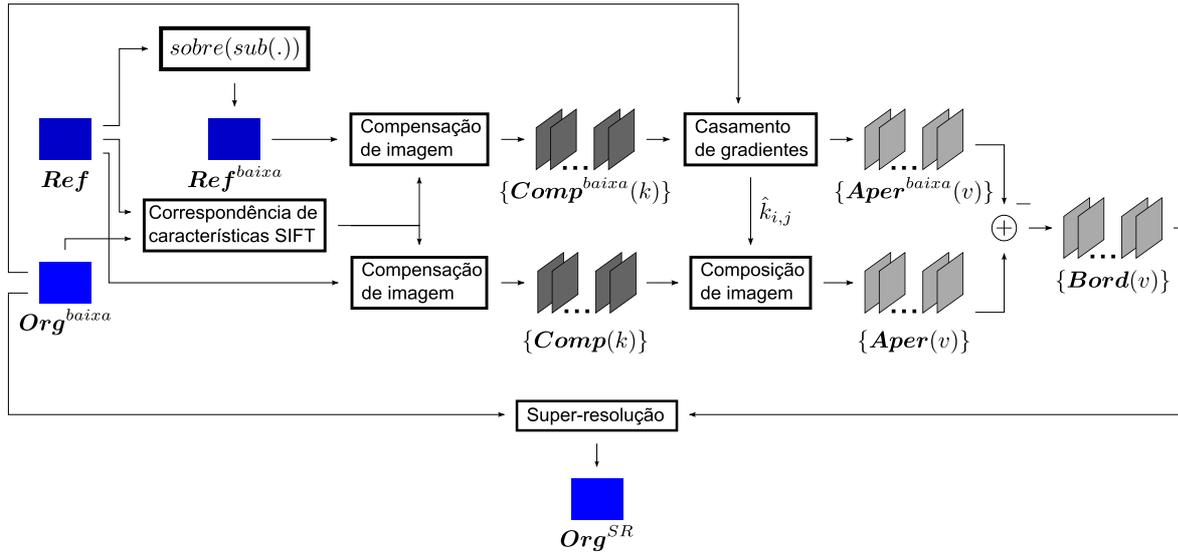


Figura 3.10: Diagrama descrevendo a combinação de técnicas aplicada a super-resolução.

é feita elemento a elemento do conjunto, com $n \in \{1, 2, \dots, N, N + 1\}$. As imagens dos conjuntos $\{Bord(v)\}_n$ são usadas para gerar a imagem $Bord^{SR}$ a ser somada à imagem Org^{baixa} , o que produzirá a imagem final super-resolvida Org^{SR} . A Figura 3.10 mostra o diagrama de como as etapas apresentadas são organizadas para a obtenção do resultado.

Uma forma de obter a imagem super-resolvida Org^{SR} é considerar simplesmente, para algum v e algum n , $Bord^{SR} = \{Bord(v)\}_n$. Uma outra maneira é realizar uma análise estatística, compondo uma única imagem $Bord^{SR}$ partir de todas as imagens dos conjuntos $\{Bord(v)\}_n$. Uma terceira e última forma que usamos obter $Bord^{SR}$ é compor um dicionário com os pares de imagens em baixa resolução e com informação em alta frequência, conforme usado por trabalhos anteriores [24, 25]. As imagens de baixa resolução do dicionário são comparadas com a imagem Org^{baixa} para a tomada de decisão de como as imagens em $\{Bord(v)\}_n$ podem ser combinadas para gerar a imagem $Bord^{SR}$. Um dicionário pode ser composto usando os conjuntos $\{Aper^{baixa}(v)\}_n$ e $\{Bord(v)\}_n$.

Os próximos capítulos irão descrever dois métodos distintos para a execução da solução proposta, chamados método de grades móveis e método de agrupamento de vetores, que serão indicados pelos acrônimos MGM e MAV, respectivamente. Cada método traz diferentes técnicas para as várias etapas aqui apresentadas, bem como os resultados experimentais obtidos.

Capítulo 4

Compensação de movimento baseada em grades móveis

O primeiro método para solução geral proposta sobre o problema apresentado no Capítulo 3, que chamamos de método de grades móveis, traz abordagens específicas para as duas etapas principais da solução. Na primeira etapa, a compensação de movimento é feita a partir de grades móveis que definem as regiões de recortes relacionadas a grupos de vetores de correspondência. Já na segunda etapa, o casamento de gradientes é feito em uma vizinhança V quadrada em torno de cada *pixel*. Apresentamos primeiramente os detalhes de cada uma das etapas. Em seguida, mostramos os resultados experimentais do uso deste método para a super-resolução de quadros de vídeos de resolução mista.

4.1 COMPENSAÇÃO DE MOVIMENTO BASEADA EM GRADES MÓVEIS

Esta primeira etapa descreve detalhadamente como obtemos as imagens compensadas dos conjuntos $\{Comp(k)\}$ e $\{Comp^{baixa}(k)\}$ a partir das imagens em alta resolução Ref e em baixa resolução interpolada Org^{baixa} . Conforme descrito anteriormente, partimos da correspondência de características (usando descritores SIFT) entre as imagens Org^{baixa} e Ref e da geração do fluxo de vetores de correspondência dos descritores, conforme exemplificado nas Figuras 3.2 e 3.4. Em seguida, fazemos o agrupamento de vetores de movimento e a definição das regiões de recorte. Neste primeiro método para o problema, primeiramente definimos as regiões de recorte e consideramos o agrupamento dos vetores internos a cada região. Esta etapa é composta dos seguintes passos, com destaque para os passos específicos deste método, que serão detalhados em seguida:

- Detecção de características SIFT nas imagens Org^{baixa} e Ref ;
- Correspondência dos descritores das características detectadas;
- Composição de um fluxo de vetores a partir da diferença de posição dos descritores correspondidos;
- **Agrupamento de vetores dentro de uma região quadrada pré-estabelecida;**
- Definição de homografias a partir dos grupos de vetores;

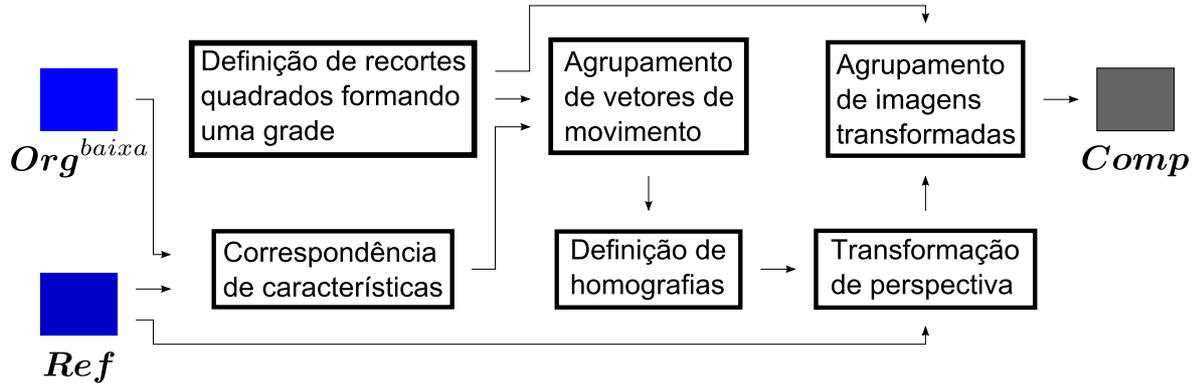


Figura 4.1: Compensação de movimento baseada em grades móveis.

- Transformação de perspectiva da imagem *Ref*;
- Composição de um mosaico pelo agrupamento de recortes das imagens transformadas.

A estrutura geral é representada pelo diagrama mostrado na Figura 4.1.

A descrição da técnica é apresentada mediante um exemplo didático conforme mostrado na Figura 4.2, que mostra a arbitragem de regiões retangulares e uma imagem *Comp* resultante. A Figura 4.2a mostra a imagem em baixa resolução *Org^{baixa}* com as regiões demarcadas sobrepostas, formando uma grade. A imagem em alta resolução *Ref* é mostrada na Figura 4.2b. Usando neste exemplo a notação apresentada no capítulo anterior (em que G representa o número de grupos de vetores), temos $G = 4$ grupos e seus respectivos recortes. Os vetores internos a cada uma das regiões demarcadas (não mostrados na figura) são usados para derivar quatro funções de transformações de perspectiva $\tau_g(\cdot)$ com $g \in \{1, 2, 3, 4\}$. As Figuras 4.2d a 4.2g mostram as versões transformadas da imagem *Ref*, ou seja, $\tau_g(\mathbf{Ref})$ com $g \in \{1, 2, 3, 4\}$. Cada máscara binária M_g usada para multiplicar sua respectiva imagem transformada é definida com valor 1 na região interna ao retângulo e 0 fora dele. Temos finalmente a imagem compensada dada por $\mathbf{Comp} = \sum_{g=1}^4 M_g \circ \tau_g\{\mathbf{Ref}\}$. Da mesma forma como já foi apresentado, uma versão aproximadamente em baixa resolução \mathbf{Comp}^{baixa} da imagem *Comp* (exceto pelas bordas das regiões de recorte retangulares) é obtida como $\mathbf{Comp}^{baixa} = \sum_{g=1}^4 M_g \circ \tau_g\{\mathbf{Ref}^{baixa}\}$, com $\mathbf{Ref}^{baixa} = \text{sobre}(\text{sub}(\mathbf{Ref}))$.

Buscando compor o par de conjuntos $\{\mathbf{Comp}(k)\}$ e $\{\mathbf{Comp}^{baixa}(k)\}$, definimos dois parâmetros relacionados à grade. O primeiro, chamado de tamanho de regiões da grade e representado por $TGrade$, define o tamanho das regiões que compõem a grade, arbitradas com forma quadrada, por questão de praticidade. Dependendo do valor do $TGrade$, a grade não engloba toda a imagem, o que causaria a super-resolução de apenas um determinado segmento da imagem. Assim, definimos o segundo parâmetro que determina o quanto o canto superior esquerdo da grade é deslocado do canto superior direito da imagem, em coordenadas horizontal e vertical. Denominamos este parâmetro de deslocamento da grade e representado por $DGrade$. A Figura 4.3 mostra exemplos de grades de diferentes $TGrade$ e $DGrade$ sobrepostas a um quadro de vídeo.

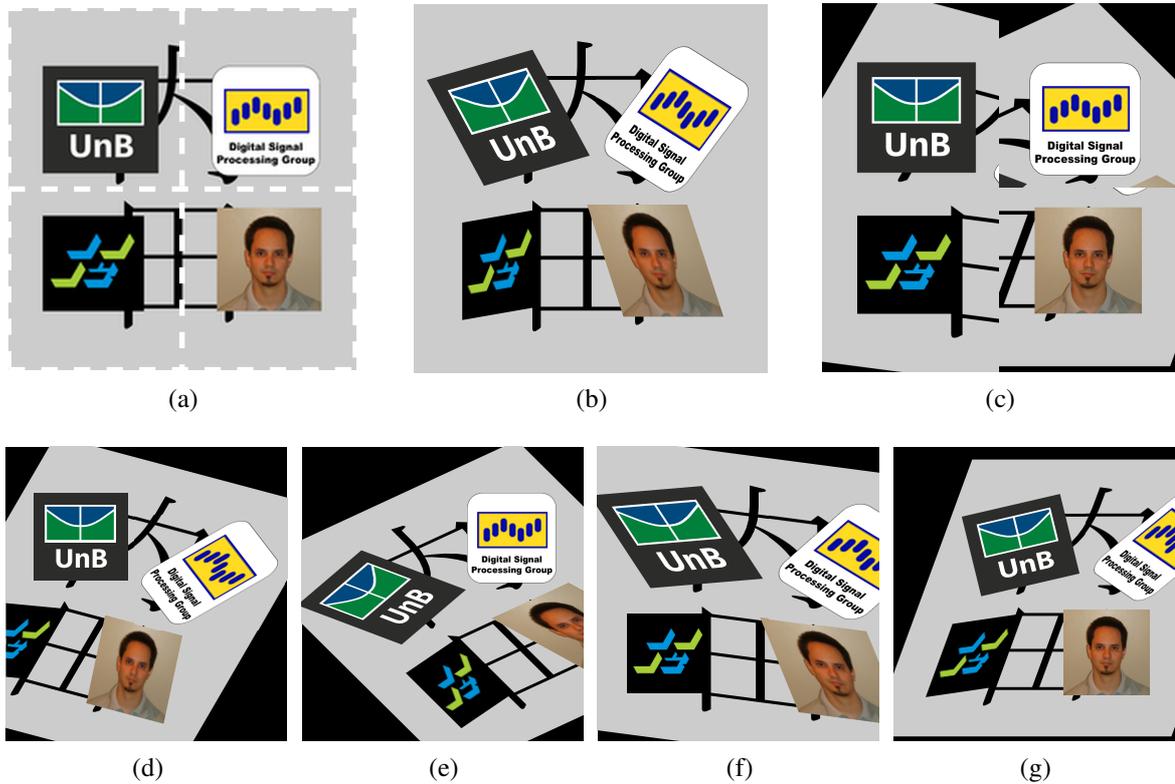


Figura 4.2: Exemplo de compensação: (a) imagem em baixa resolução interpolada Org^{baixa} com grade sobreposta; (b) imagem em alta resolução Ref ; (c) imagem compensada $Comp$. Imagens transformadas: (d) $\tau_1(Ref)$; (e) $\tau_2(Ref)$; (f) $\tau_3(Ref)$; (g) $\tau_4(Ref)$.

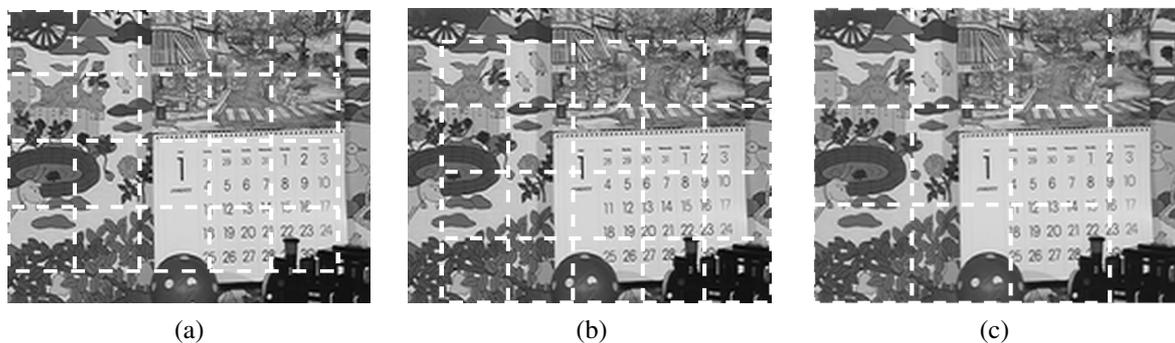


Figura 4.3: Exemplos de grades com $TGrade = 64$ pixels e (a) $DGrade = (0, 0)$, (b) $DGrade = (32, 32)$; (c) $TGrade = 96$ pixels e $DGrade = (0, 0)$. Os deslocamentos, em pixels, são dados nas direções horizontal e vertical, respectivamente.

A partir da variação dos parâmetro $TGrade$ e $DGrade$ somos capazes de gerar diversas imagens compensadas. A indexação k do conjunto $\{Comp(k)\}$ não obedece a nenhuma regra específica relacionada aos parâmetros, é simplesmente uma enumeração das imagens compensadas obtidas. Assim, podemos dizer que k define uma grade específica determinada pelos dois parâmetros e, para simplificar a notação, vamos omiti-lo doravante. A Figura 4.4 traz um exemplo da composição do conjunto $\{Comp\}$ usando $TGrade = 128\ pixels$ e $DGrade$ em passos de $32\ pixels$ em ambas as direções.

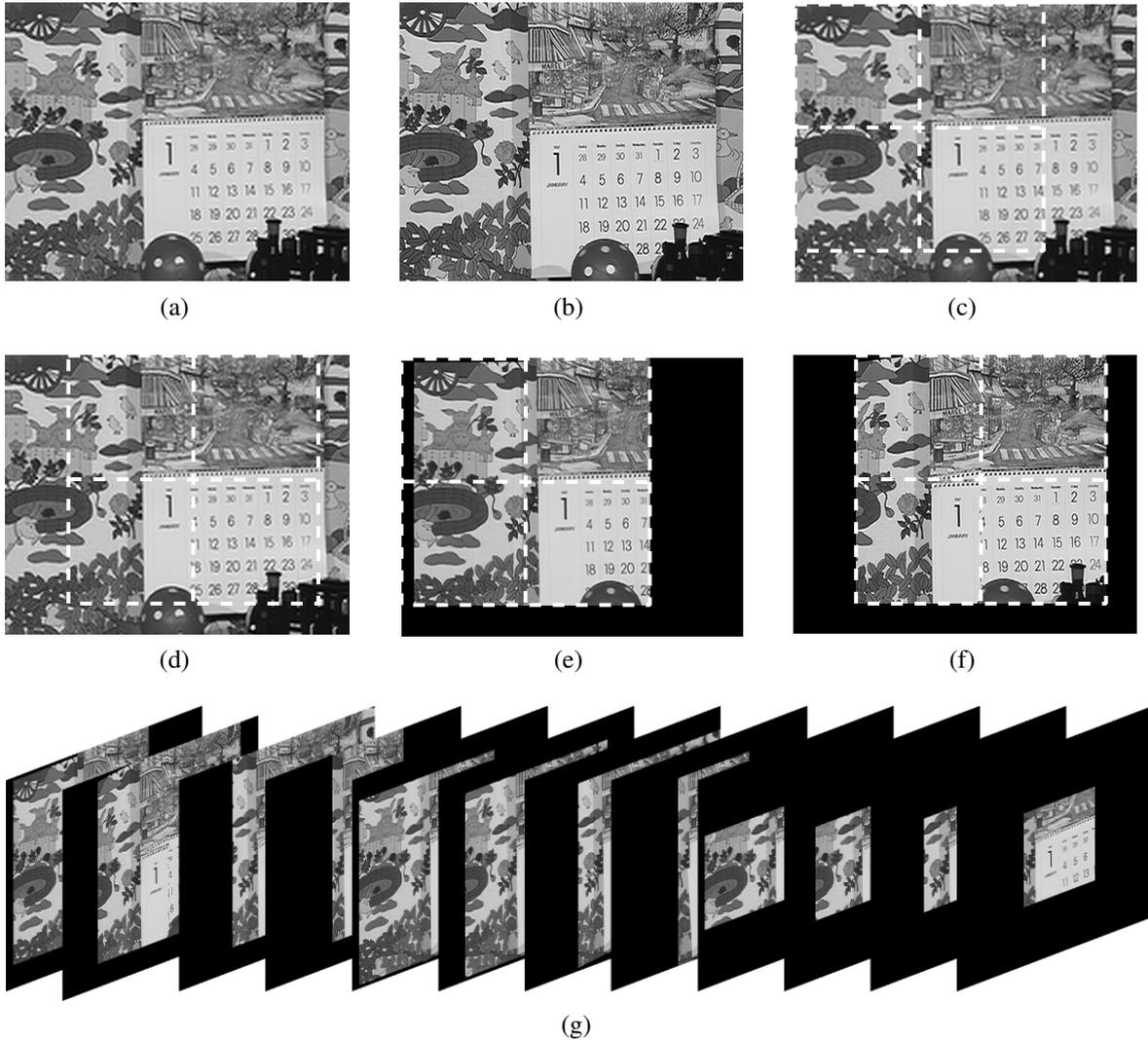


Figura 4.4: Exemplo do conjunto $\{Comp(k)\}$ usando $TGrade = 128\ pixels$: (a) imagem Org^{baixa} ; (b) imagem Ref ; imagem Org^{baixa} com grades (c) $k = 1$ e (d) $k = 2$ sobreposta; (e) $Comp(1)$; (f) $Comp(2)$; (g) conjunto de imagens $\{Comp(k)\}$.

4.2 CASAMENTO DE GRADIENTES EM VIZINHANÇAS QUADRADAS

Conforme já apresentado, a compensação de movimento produz imagens $\{Comp^{baixa}\}$ semelhantes à imagem Org^{baixa} que desejamos super-resolver, bem como suas versões em alta resolução $\{Comp\}$. Contudo, a obtenção de informação de alta frequência pode ser melhorada usando casamento de gradientes. Buscamos então gerar o par de conjuntos $\{Aper^{baixa}(v)\}$ e $\{Aper(v)\}$ dos quais obteremos a informação de alta frequência $Bord^{SR}$.

Foi apresentado o casamento de gradientes considerando a vizinhança em torno de cada *pixel*. Neste primeiro método proposto, nós fazemos o casamento de gradientes considerando uma vizinhança quadrada em torno de cada *pixel*. O casamento é feito entre os gradientes das imagens Org^{baixa} e $\{Comp^{baixa}\}$. Então, para $Org_{i,j}^{baixa}$ um *pixel* na imagem Org^{baixa} e $Comp_{i,j}^{baixa}$ um *pixel* na imagem $Comp^{baixa}$ e $\nabla Org_{i,j}^{baixa}$ e $\nabla Comp_{i,j}^{baixa}$ seus respectivos gradientes, buscamos o índice $\hat{k}_{i,j}$ que satisfaça

$$\hat{k}_{i,j} = \underset{k}{\operatorname{argmin}} \sum_{s=-v}^v \sum_{t=-v}^v \|\nabla Org_{i+v,j+t}^{baixa} - \nabla Comp_{i+v,j+t}\|, \quad (4.1)$$

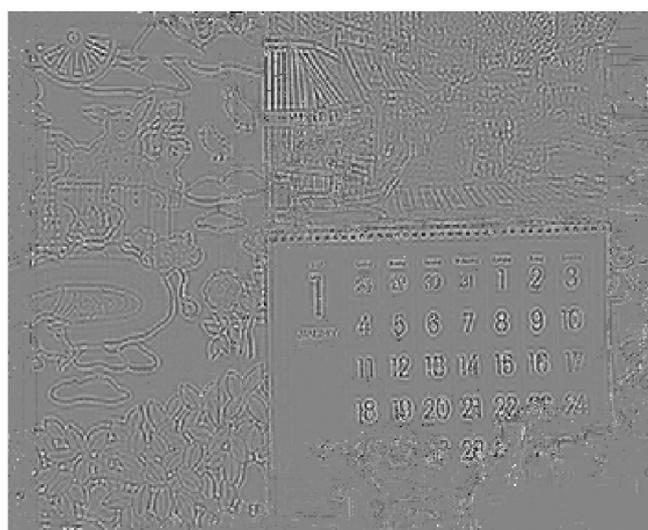
sendo a janela quadrada de busca em torno de cada *pixel* com tamanho $2v + 1$ e os gradientes calculados com operador de Sobel. Por determinar o tamanho da janela, referimo-nos ao índice v como tamanho da vizinhança e o representamos por $TViz$. Assim, para cada valor de v , compomos uma imagem $Aper^{baixa}(v)$ cujos *pixels* são $Aper_{i,j}^{baixa}(v) = Comp_{i,j}^{baixa}(\hat{k}_{i,j})$. Consequentemente, temos também a imagem $Aper(v)$, cujos *pixels* são dados por $Aper_{i,j}(v) = Comp_{i,j}(\hat{k}_{i,j})$. Assim como no modelo geral, as imagens contendo apenas informação de alta frequência são obtidas por $Bord(v) = Aper(v) - Aper^{baixa}(v)$. As Figuras 4.5 a 4.7 mostram exemplos de imagens aprimoradas em alta e baixa resolução, bem como a informação de alta frequência para diferentes valores de $TViz$. Podemos observar com bastante clareza que o refinamento é aprimorado com o uso de uma vizinhança.



(a)



(b)



(c)

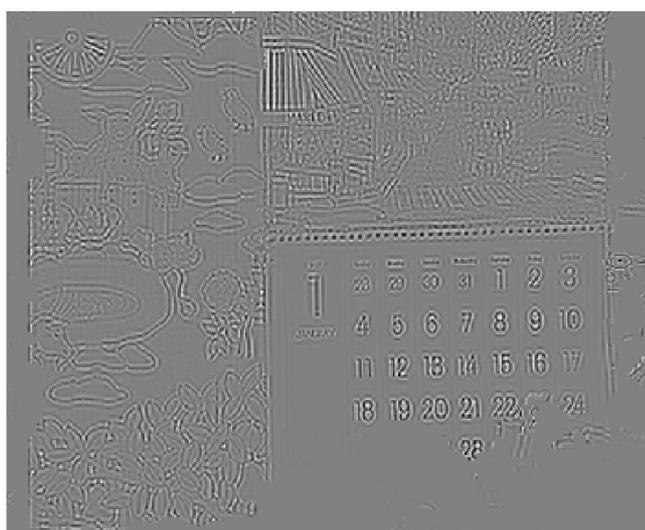
Figura 4.5: Exemplo de imagens aprimoradas após o casamento de gradientes para $TVIz = 0$: (a) $Aper(0)$, (b) $Aper^{baixa}(0)$ e (c) $Bord(0)$.



(a)



(b)



(c)

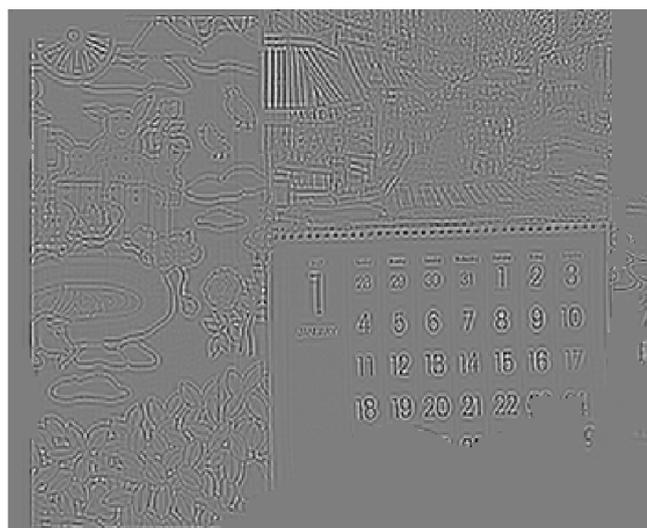
Figura 4.6: Exemplo de imagens aprimoradas após o casamento de gradientes para $TVIz = 7$: (a) $Aper(7)$, (b) $Aper^{baixa}(7)$ e (c) $Bord(7)$.



(a)



(b)



(c)

Figura 4.7: Exemplo de imagens aprimoradas após o casamento de gradientes para $TVIz = 15$: (a) $Aper(15)$, (b) $Aper^{baixa}(15)$ e (c) $Bord(15)$.

4.3 SUPER-RESOLUÇÃO DE QUADROS DE VÍDEO

Os primeiros experimentos visando avaliar nosso método foram executados em quadros de vídeos não comprimidos, seguindo as mesmas condições de trabalhos anteriores [24, 25], o que permite uma comparação direta de resultados. As condições foram as seguintes:

- A imagem a ser super-resolvida (quadro-não-chave) é uma versão subamostrada do 16º quadro original;
- As imagens usadas como referência (quadros-chave) são o 1º (quadro 0 da sequência) e o 31º (quadro 30 da sequência) quadros originais;
- Em ambas as etapas de subamostragem e sobreamostragem, representadas pelas funções *sub(.)* e *sobre(.)*, respectivamente, o fator de escala é 2 e o filtro usado é o Lanczos-3 (usado em [25]);
- Foram testadas quatro sequências de tamanho CIF (*container*, *hall*, *mobile* e *news*) e duas sequências de tamanho 720p (*mobcal* e *shields*);
- A métrica usada para avaliação objetiva dos resultados foi a PSNR.

Em [24] é apresentado um filtro *anti-aliasing* apenas por seus coeficientes, não sendo especificando o tipo de filtro. Os quadros usados nos testes são mostrados nas Figuras 4.8 a 4.13.

Com relação a parâmetros específicos do nosso método, testamos os seguintes valores, arbitrados para avaliações iniciais.

- Para as sequências de tamanho CIF, valores para o parâmetro *TGrade* iguais a 64, 128 e 256.
- Para as sequências de tamanho 720p, valores para o parâmetro *TGrade* iguais a 128, 256 e 512.
- Valores para o parâmetro *DGrade* iguais a múltiplos de 1/16 do valor do parâmetro *TGrade*. Por exemplo, para uma grade com *DGrade* = 128, o seu deslocamento se dá em passos de 8 em 8 *pixels* nas direções horizontal e vertical.

O deslocamento da grade em saltos proporcionais ao tamanho das regiões visam ao agrupamento de diferentes conjuntos de vetores do fluxo obtido pela correspondência de descritores. Nós primeiramente executamos testes buscando averiguar melhores condições e parâmetros para apenas posteriormente compararmos nossos resultados com aqueles dos trabalhos anteriores.



Figura 4.8: Quadros da sequência *container* usados nos testes: (a) 1º quadro original, (b) 16º quadro reamostrado e (c) 31º quadro original.

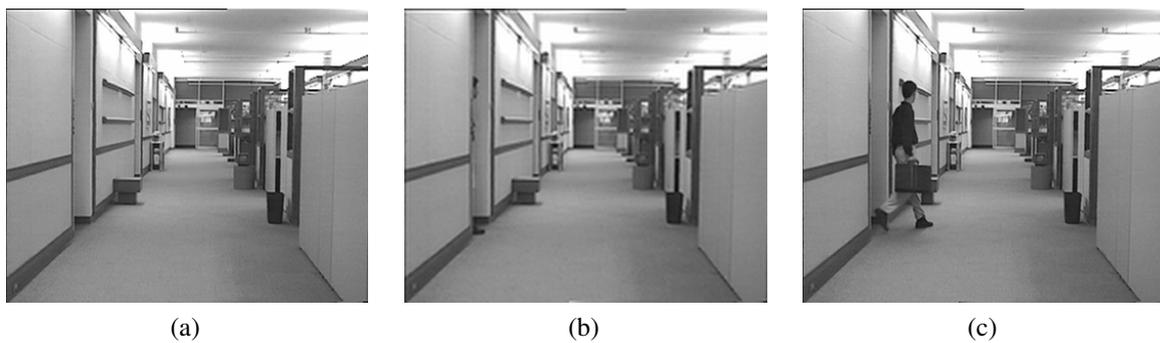


Figura 4.9: Quadros da sequência *hall* usados nos testes: (a) 1º quadro original, (b) 16º quadro reamostrado e (c) 31º quadro original.

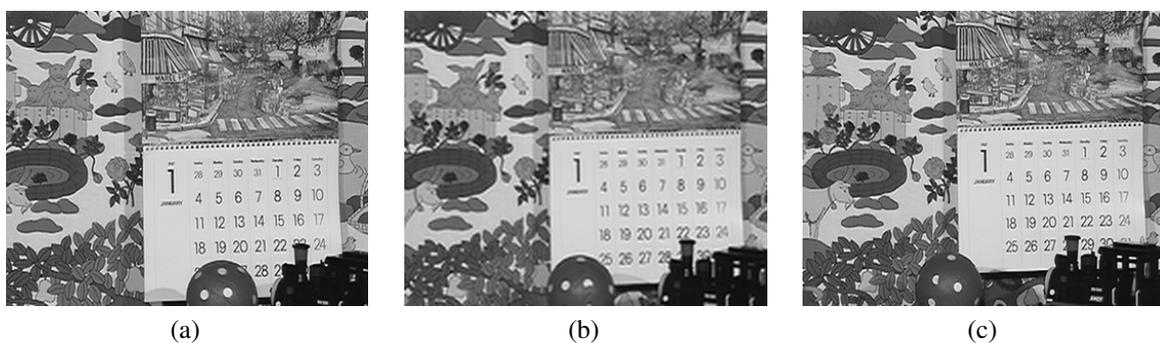


Figura 4.10: Quadros da sequência *mobile* usados nos testes: (a) 1º quadro original, (b) 16º quadro reamostrado e (c) 31º quadro original.



Figura 4.11: Quadros da sequência *news* usados nos testes: (a) 1º quadro original, (b) 16º quadro reamostrado e (c) 31º quadro original.

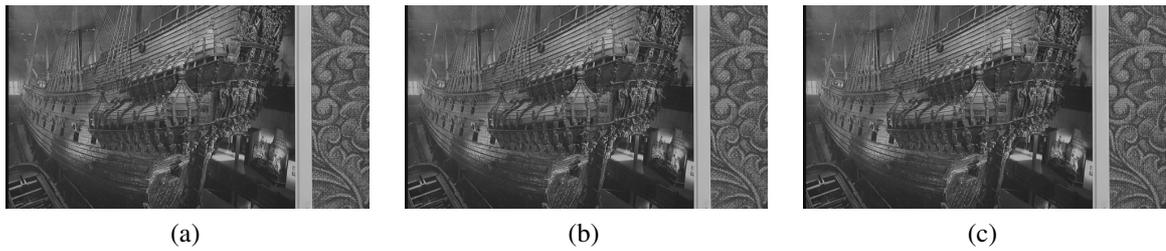


Figura 4.12: Quadros da sequência *mobcal* usados nos testes: (a) 1º quadro original, (b) 16º quadro reamostrado e (c) 31º quadro original.

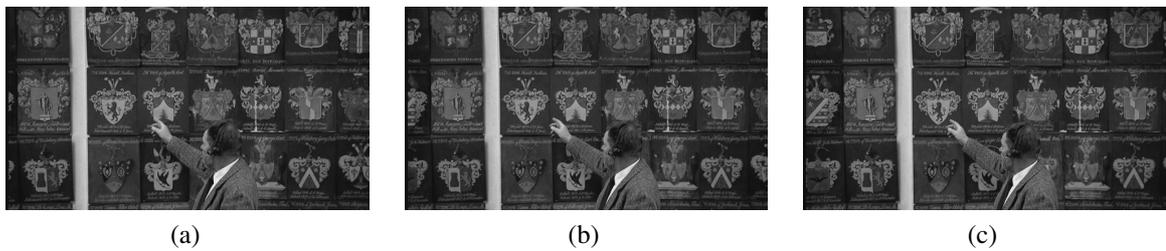


Figura 4.13: Quadros da sequência *shields* usados nos testes: (a) 1º quadro original, (b) 16º quadro reamostrado e (c) 31º quadro original.

4.4 ADIÇÃO DIRETA DE INFORMAÇÃO DE ALTA FREQUÊNCIA

O primeiro teste realizado para avaliar nosso método foi a adição direta da informação de alta frequência $Bord(v)$ ao quadro não-chave Org^{baixa} de baixa resolução interpolada, ou seja, o quadro super-resolvido A^{SR} é obtido por $A^{SR} = Org^{baixa} + Bord(v)$, para diferentes imagens $Bord(v)$. Neste primeiro teste, buscamos averiguar a influência de três questões principais na super-resolução: a influência do uso de um ou dois quadros-chave de referência Ref_n , com $n \in \{1, 2\}$; a influência dos valores do parâmetro de tamanho de grade $TGrade$ na etapa de compensação de movimento; a influência do tamanho da vizinhança quadrada $TViz$ na etapa de casamento de gradientes.

Para os diferentes usos de quadros de referência, temos que Ref_1 é o 1º enquanto Ref_2 é 31º quadro da sequência. Conforme apresentado, os dois quadros Ref_n levam aos pares de conjuntos $\{Comp\}_n$ e $\{Comp^{baixa}\}_n$, o que leva aos conjuntos $\{Aper(v)\}_n$ e $\{Aper^{baixa}(v)\}_n$. Lembramos que é possível usar, na etapa de casamento de gradientes, o conjunto $\{Comp^{baixa}\}_3 = \{Comp^{baixa}\}_1 \cup \{Comp^{baixa}\}_2$ o que resulta nos conjuntos $\{Aper(v)\}_3$ e $\{Aper^{baixa}(v)\}_3$ e consequentemente $\{Bord(v)\}_3$.

A variação do parâmetro $TGrade$, por assumir valores distintos para diferentes tamanhos de vídeos, é mencionada nos resultados pelos termos *Pequeno*, *Médio* e *Grande* para nos referirmos aos valores 64, 128 e 256 para as sequências de tamanho CIF e 128, 256 e 512 para sequências de tamanho 720p, respectivamente. Usamos as notações $\{Comp_P\}_n$, $\{Comp_M\}_n$ e $\{Comp_G\}_n$ para simbolizar os conjuntos resultantes do uso dos tamanhos *Pequeno*, *Médio* e *Grande*, respectivamente. Temos então um total de nove conjuntos $\{Comp_{TGrade}\}_n$ para cada condição de teste (e seus respectivos $\{Comp_{TGrade}^{baixa}\}_n$), o que dá origem a nove distintos conjuntos $\{Bord_{TGrade}(v)\}_n$, indicados por $\{Bord_P\}_n$, $\{Bord_M\}_n$ e $\{Bord_G\}_n$, para $n \in \{1, 2, 3\}$.

Para o parâmetro $TViz$, testamos valores variando de 0 a 15. O valor máximo 15 foi arbitrado por produzir uma janela de 31×31 pixels, ou seja, aproximadamente metade das dimensões do menor valor de $TGrade$.

Os resultados obtidos são mostrados nas Figuras 4.14 a 4.19. Cada gráfico mostra os diferentes usos de quadros-chave e a variação de $TViz$ para cada tamanho $TGrade$. Os valores apresentados são cálculos de PSNR entre o quadro-não-chave super-resolvido e sua versão original.

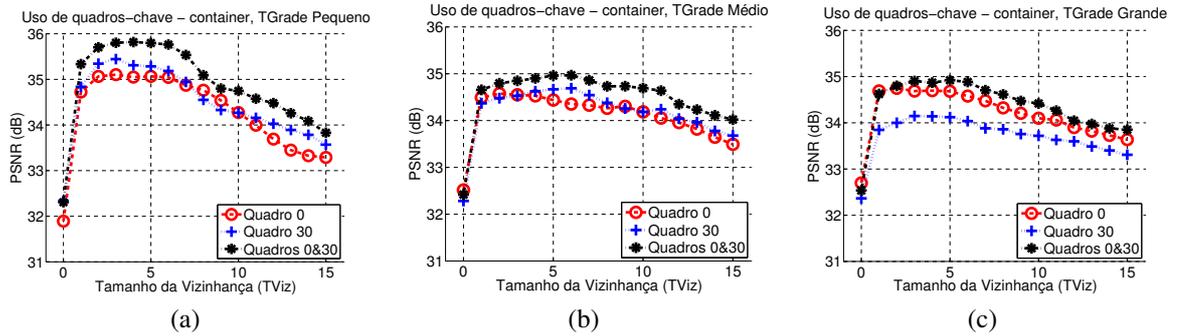


Figura 4.14: Comparação de valores de PSNR entre diferentes usos de quadros-chave, variando TViz para seqüência *container* e tamanhos de blocos: (a) Pequenos, (b) Médios e (c) Grandes.

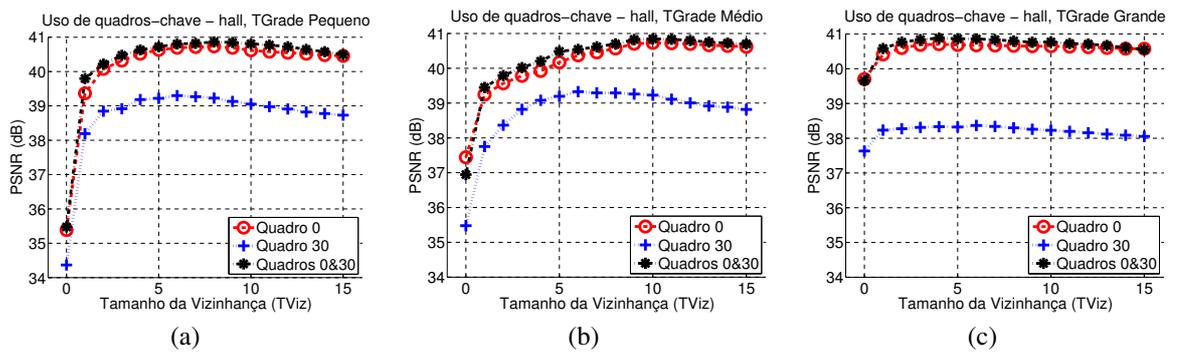


Figura 4.15: Comparação de valores de PSNR entre diferentes usos de quadros-chave, variando TViz para seqüência *hall* e tamanhos de blocos: (a) Pequenos, (b) Médios e (c) Grandes.

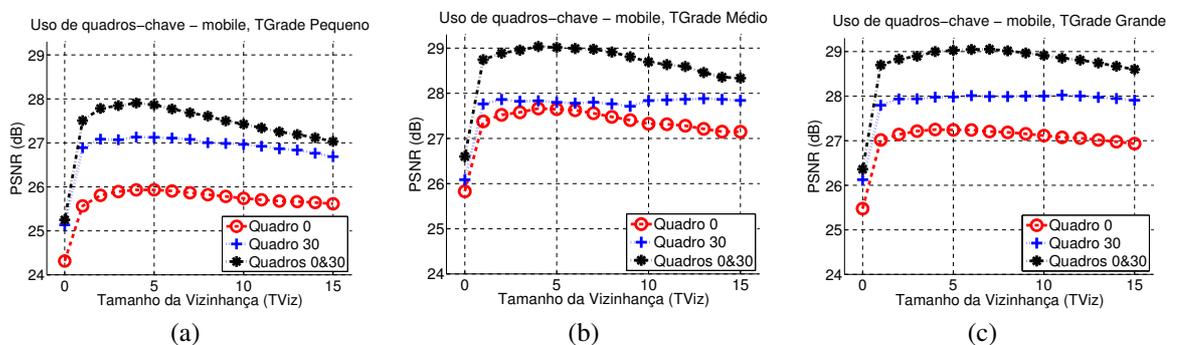


Figura 4.16: Comparação de valores de PSNR entre diferentes usos de quadros-chave, variando TViz para seqüência *mobile* e tamanhos de blocos: (a) Pequenos, (b) Médios e (c) Grandes.

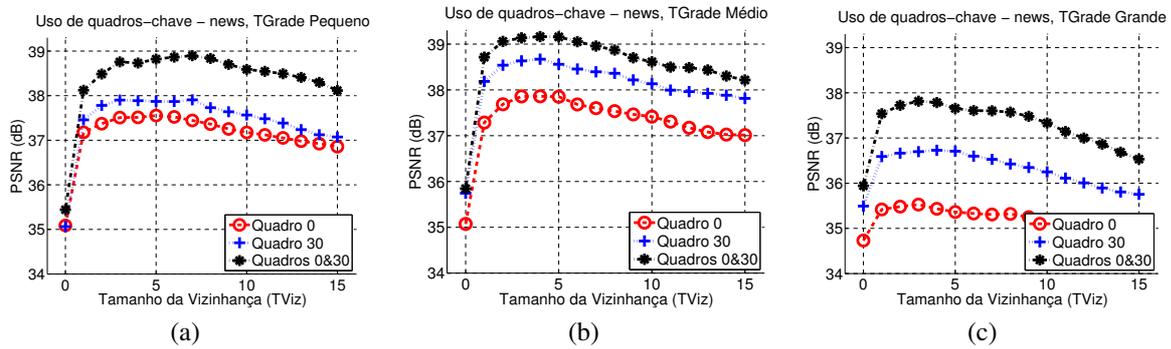


Figura 4.17: Comparação de valores de PSNR entre diferentes usos de quadros-chave, variando TViz para sequência *news* e tamanhos de blocos: (a) Pequenos, (b) Médios e (c) Grandes.

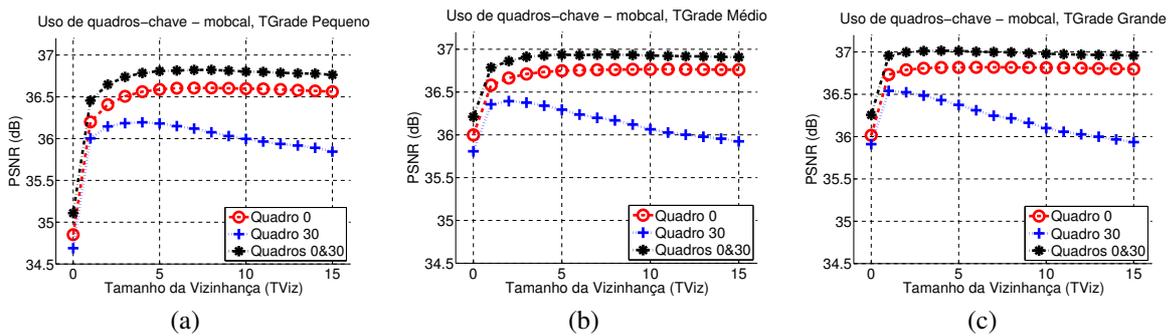


Figura 4.18: Comparação de valores de PSNR entre diferentes usos de quadros-chave, variando TViz para sequência *mobcal* e tamanhos de blocos: (a) Pequenos, (b) Médios e (c) Grandes.

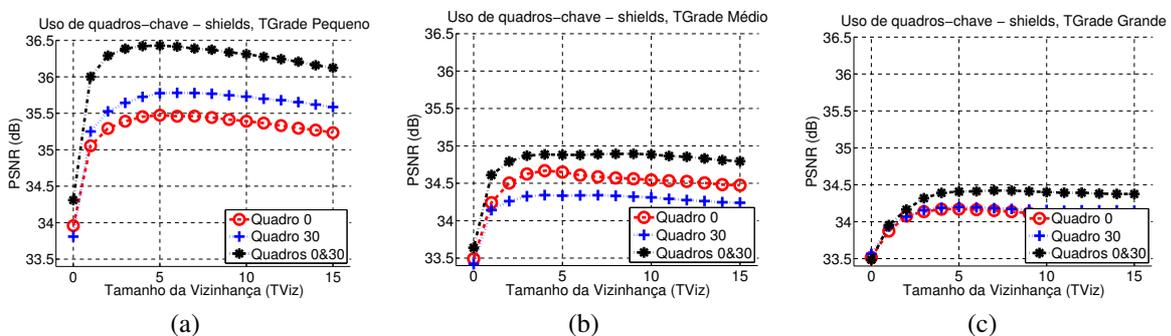


Figura 4.19: Comparação de valores de PSNR entre diferentes usos de quadros-chave, variando TViz para sequência *shields* e tamanhos de blocos: (a) Pequenos, (b) Médios e (c) Grandes.

Observando os gráficos, fica muito claro notar que, para absolutamente todos os casos, usar dois quadros-chave como referência é melhor que usar apenas um. Isso é mostrado na Tabela 4.1. Essa tabela mostra a diferença de PSNR entre o uso de dois quadros-chave e apenas um (o melhor dos casos individualmente) para cada sequência e tamanho $TGrade$. Podemos observar da Tabela que em alguns casos a vantagem em usar dois quadros não é muito grande, como no caso das sequências *hall* e *mobcal*. Isso ocorre porque essas são sequências com pouco movimento entre um dos quadros-chave e o quadro-não-chave. Já nas sequências com mais movimento, em especial a sequência *mobile*, o ganho pelo uso de dois quadros é maior.

Observamos também que existe uma faixa intermediária de valores de $TViz$ com resultados superiores aos demais. Notado isso, podemos procurar agora qual o valor de $TViz$ que, na média, leva aos melhores resultados. Observando os gráficos, notamos claramente o quanto o resultado obtido pelo uso de $TViz = 0$ é consideravelmente menor que os demais. Por isso, consideramos nesta análise apenas os valores de $TViz$ tal que $1 \leq TViz \leq 15$.

A metodologia adotada foi uma análise estatística de qual valor $TViz$ proporciona maiores valores médios de PSNR. Para uma comparação mais completa e não enviesada, visto que cada resultado obtido está dentro de faixas distintas de valores de PSNR, cada vetor de resultados dentro de uma mesma condição de testes foi normalizado. Em outras palavras, para cada curva dos gráficos mostrados nas Figuras 4.14a a 4.19c, os valores de PSNR foram agrupados em um vetor de 15 elementos e normalizados para ter seus valores indo de 0 a 1. Temos então um total de 54 vetores, sendo 9 por sequência (3 por teste de $TGrade$ e 3 por teste de uso de quadro-chave). A Figura 4.20 mostra todas as curvas dos gráficos anteriores, mas com valores normalizados.

A normalização dos vetores nos permite agora analisar estatisticamente os valores de PSNR para cada valor de $TViz$, calculando os valores de tendência central de média. Isso nos permite verificar qual $TViz$ tem uma maior concentração de valores próximos a 1. A Figura 4.21 mostra os histogramas dos valores de PSNR normalizados apresentados na Figura 4.20, para cada valor de $TViz$, bem como os valores de média para cada um deles. Observando que os valores de média mais elevados ocorrem quando $TViz = 5$, concluímos finalmente: este é o valor que, na média, leva aos melhores resultados.

Por fim, analisando os resultados para diferentes tamanhos $TGrade$, percebemos que

Tabela 4.1: Diferença de PSNR [dB] entre o uso de dois e um quadros-chave como referência

Sequência	Tamanho dos blocos		
	Pequeno	Médio	Grande
<i>container</i>	0.4	0.3	0.2
<i>hall</i>	0.1	0.1	0.2
<i>mobile</i>	0.8	1.2	1.0
<i>news</i>	1.0	0.5	1.1
<i>mobcal</i>	0.2	0.2	0.2
<i>shields</i>	0.6	0.2	0.2

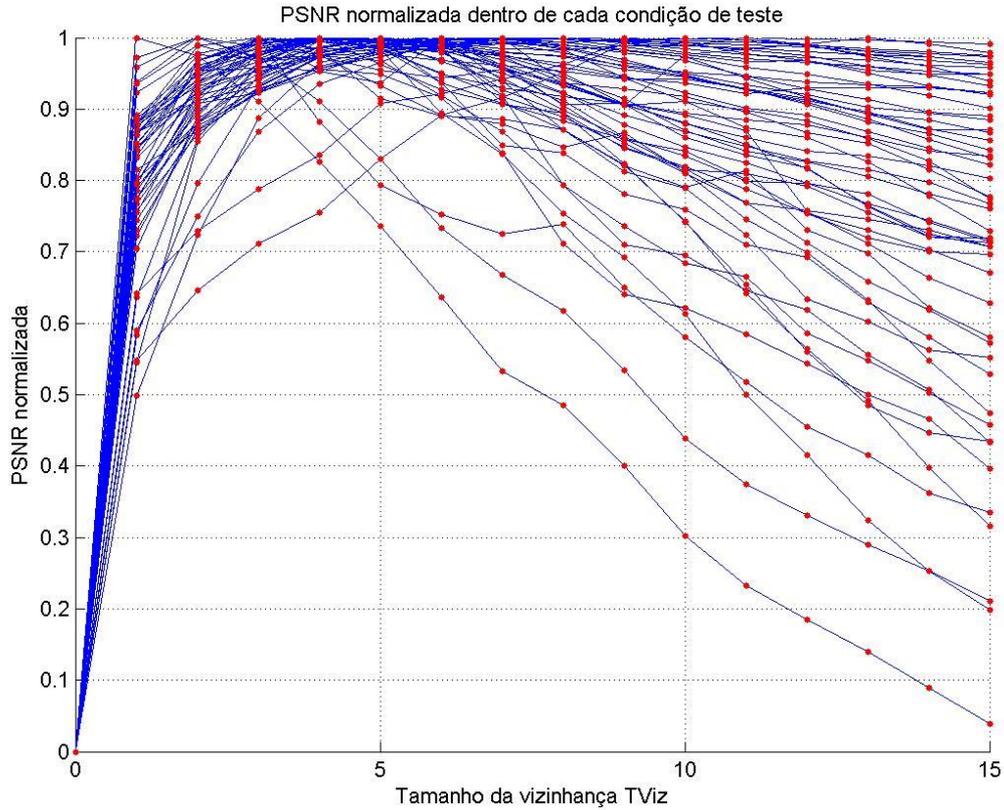


Figura 4.20: Curvas de PSNR dos gráficos mostrados nas Figuras 4.14a a 4.19c normalizadas, onde buscamos observar qual valor de $TViz$ concentra mais pontos próximos a 1.

há grande variedade dependendo da sequência analisada. Por exemplo, para as sequências *container* e *shields*, o uso de regiões com $TGrade$ assumindo valor *Pequeno* tem desempenho consideravelmente superior aos demais (superior a $1dB$), ao passo que para a sequência *mobile* os tamanhos *Médio* e *Grande* têm desempenho superior aproximado, enquanto que para a sequência *news* o desempenho superior aproximado se dá para os tamanhos *Pequeno* e *Médio*. Por isso, não é possível definir qual dos tamanhos *Pequeno*, *Médio* ou *Grande* é melhor. Isso nos leva aos próximos testes, em que buscamos usar todas as imagens $Bord(v)$ disponíveis contendo informação de alta frequência para o cálculo da imagem super-resolvida.

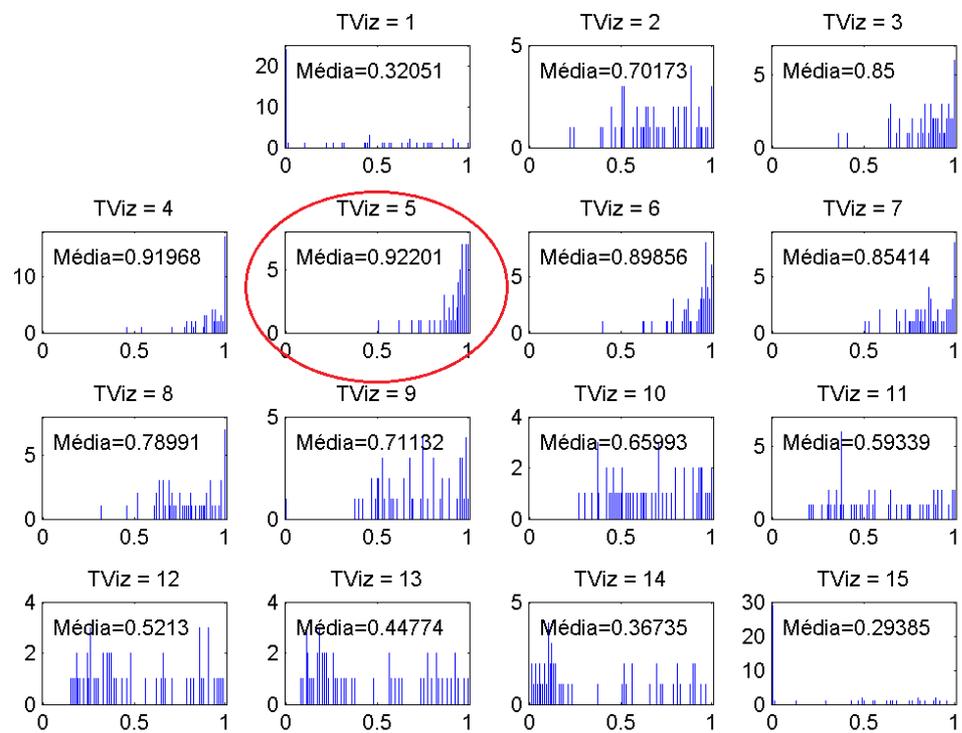


Figura 4.21: Histogramas dos valores de PSNR normalizados mostrados nas Figuras 4.20, para cada valor de $TViz$, referentes à SR por adição direta.

4.5 SUPER-RESOLUÇÃO POR ANÁLISE ESTATÍSTICA DA INFORMAÇÃO DE ALTA FREQUÊNCIA

Conforme apresentado no capítulo 3, podemos obter a informação final de alta frequência a ser adicionada ao quadro-não-chave por análise estatística das imagens nos conjuntos $\{\mathbf{Bord}_{TGrade}(v)\}_n$. Lembrando, variamos $TGrade$ com tamanhos *Pequeno*, *Médio* ou *Grande*, $n \in 1, 2, 3$ e $TViz$ representado pelo índice v com valores $1 \leq v \leq 15$. A única diferença entre as imagens geradas para este teste e o teste anterior foi o uso do parâmetro de deslocamento da grade $DGrade$ igual a $1/4$ do parâmetro do tamanho de grade $TGrade$, em vez de $1/16$. Testes preliminares mostraram que essa mudança praticamente não traz diferenças nos resultados objetivos, ao passo que reduz a complexidade da etapa de casamento de gradientes por gerar conjuntos $\{\mathbf{Comp}_{TGrade}\}_n$ menores. Isso se explica pelo fato de que passos muito pequenos no deslocamento da grade podem fazer com que diferentes regiões de recorte agrupem os mesmos vetores, gerando as mesmas imagens transformadas, o que não agrega nenhuma informação extra para a super-resolução. Assim, para realizar a análise estatística, unimos os conjuntos $\{\mathbf{Bord}_{TGrade}(v)\}_n$, para todos os valores de $TGrade$ e n e para todas as vizinhanças de 1 a $TViz$, num conjunto único $\{\mathbf{Bord}(\nu)\}^v$:

$$\{\mathbf{Bord}(\nu)\}^{TViz} = \bigcup_{n \in \{1,2,3\}} \bigcup_{TGrade \in \{P,M,G\}} \bigcup_{v'=1}^{TViz} \mathbf{Bord}_{TGrade}(v')_n \quad (4.2)$$

Neste teste buscamos avaliar como a composição do conjunto $\{\mathbf{Bord}(\nu)\}^{TViz}$ influencia no resultado final, para diferentes valores de $TViz$. Cada valor de $TViz$ gera um conjunto acumulativo de nove imagens, sendo três valores de $TGrade$ (indicados por P, M e G) e três valores de n . Assim, $\{\mathbf{Bord}(\nu)\}^1$ tem apenas nove imagens. Conforme aumentamos o valor de $TViz$, novos grupos de nove imagens são unidos ao grupo anterior, de forma que $\{\mathbf{Bord}(\nu)\}^2$ tem 18 elementos, $\{\mathbf{Bord}(\nu)\}^3$ tem 27 e assim por diante.

A análise estatística é feita da seguinte forma. Seja $i_{i,j}$ um vetor contendo todos os *pixels* na posição i, j de cada imagem $\{\mathbf{Bord}(\nu)\}^{TViz}$. Compomos uma nova imagem $\mathbf{Bord}^{AE}(TViz)$ (*AE* se refere a análise estatística) cujos *pixels* $Bord_{i,j}^{AE}$ são dados pela média aritmética das componentes de cada vetor $i_{i,j}$. A imagem super-resolvida final é calculada por $\mathbf{A}^{SR} = \mathbf{Org}^{baixa} + \mathbf{Bord}^{AE}(TViz)$. A Figura 4.22 mostra a geração da imagem $\mathbf{Bord}^{AE}(TViz)$ a partir do conjunto $\{\mathbf{Bord}(\nu)\}^{TViz}$.

Os resultados para este teste são mostrados em curvas de PSNR em que cada imagem \mathbf{A}^{SR} é gerada pelo uso de um valor de $TViz$ distinto, nas Figuras 4.23a a 4.23f. Os gráficos mostram também o melhor resultado obtido pela adição direta de informação de alta frequência, descrita anteriormente. Percebemos dos gráficos que, a partir de certo valor de $TViz$, o agrupamento de mais imagens passa a ser prejudicial. Isso se dá pelo fato de que uma vizinhança muito grande pode usar gradientes de regiões de recortes vizinhas àquelas do *pixel* sendo casado. No caso específico da sequência *hall*, isso não é observado por ela

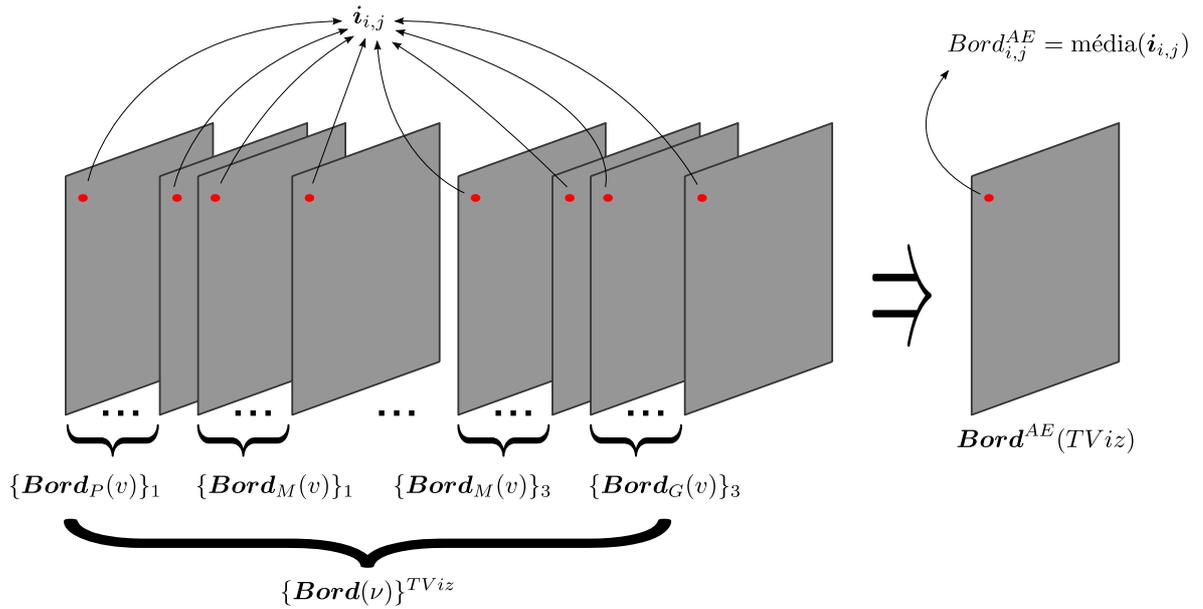


Figura 4.22: Geração da imagem com informação de alta frequência $Bord^{AE}$ a partir da média dos *pixels* colocalizados das imagens do conjunto $Bord$ composto pela união de todos os conjuntos $\{Bord_{TGrade}(v)\}_n$.

ser uma seqüência em que a câmera está parada e há muito pouco movimento na cena.

Buscamos também avaliar qual o valor de $TViz$ que produz os melhores resultados, na média. Fazendo uma análise semelhante àquela feita anteriormente, normalizamos as curvas de PSNR para cada seqüência para termos valores de zero a um, como pode ser visto na Figura 4.24. Observando o histograma dos valores normalizados para cada valor de $TViz$, podemos concluir que, na média, $TViz = 9$ produz o melhor resultado.

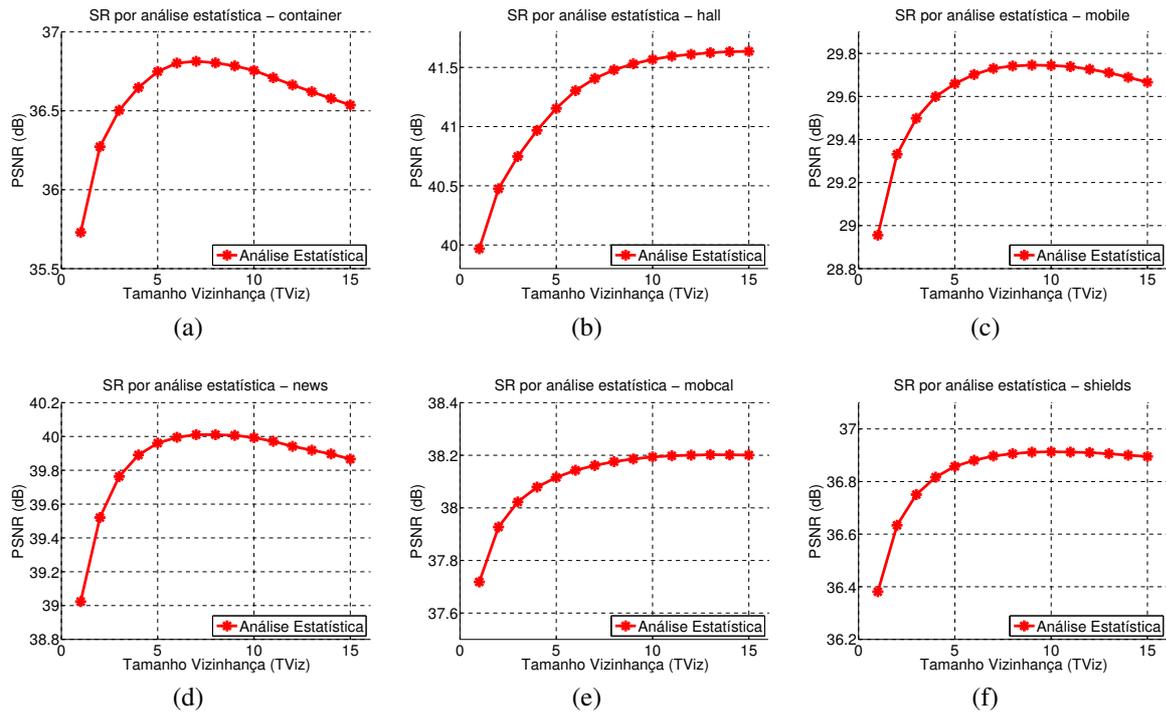


Figura 4.23: Comparação de valores de PSNR resultantes da composição de conjuntos para diferentes valores de tamanho da vizinhança $TViz$, para todas as seqüências: (a) *container*; (b) *hall*; (c) *mobile*; (d) *news*; (e) *mobcal*; (f) *shields*.

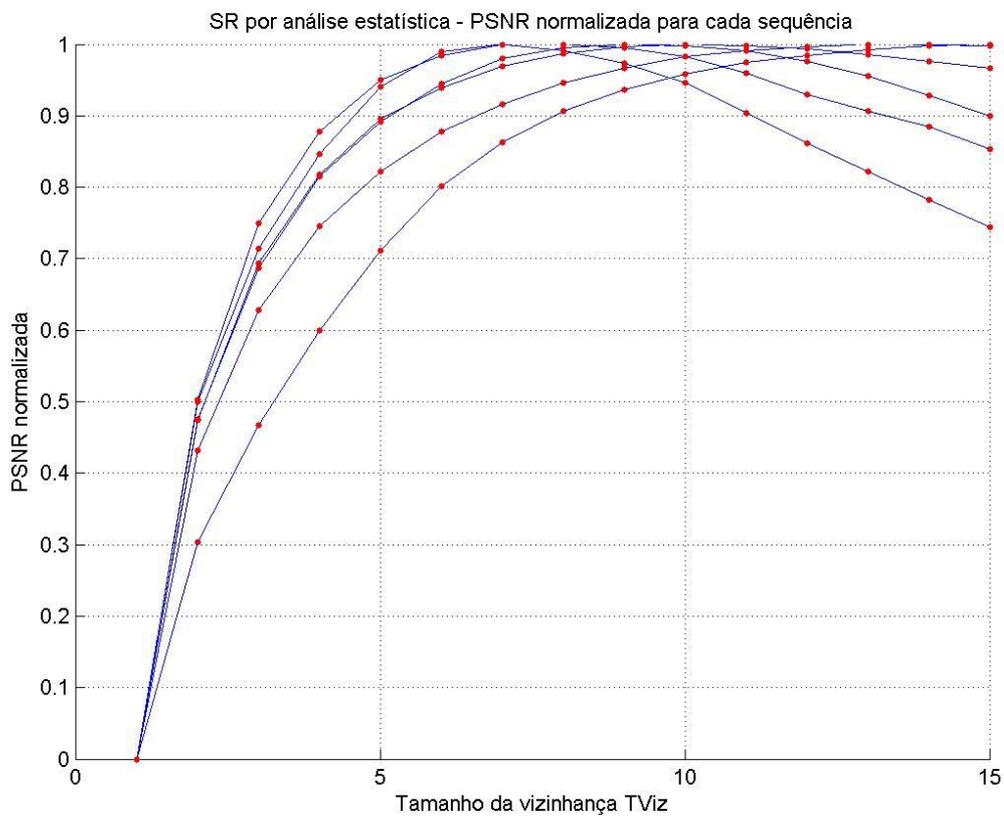


Figura 4.24: Curvas de PSNR dos gráficos mostrados nas Figuras 4.23a a 4.23f normalizadas.

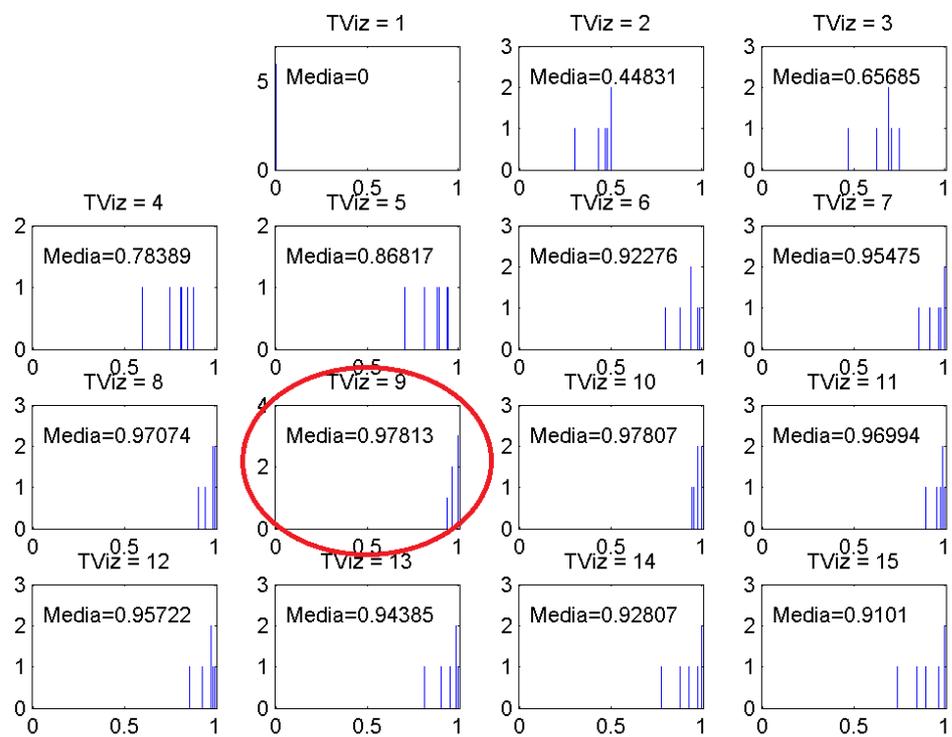


Figura 4.25: Histograma dos valores de PSNR normalizados mostrados na Figura 4.24, para cada valor de $TViz$, referentes à SR por análise estatística.

4.6 SUPER-RESOLUÇÃO POR PONDERAÇÃO DE DICIONÁRIO

No Capítulo 3, apresentamos três formas de realizar a super-resolução a partir das imagens contendo informação de alta frequência $\mathbf{Bord}(v)$ obtidas pelo método que nós propomos. As duas primeiras formas foram apresentadas nas seções anteriores. Mostramos agora como usamos a técnica proposta por Hung *et al.* [25] (apresentada no Capítulo 2) para alcançarmos melhores resultados.

Na seção anterior mostramos que podemos compor diversos conjuntos $\{\mathbf{Bord}(\nu)\}^{TViz}$ a partir da união de conjuntos $\{\mathbf{Bord}_{TGrade}(v)\}_n$ obtidos para distintas condições de teste. Lembramos também que cada imagem $\mathbf{Bord}_{TGrade}(v)$ é obtida por $\mathbf{Bord}_{TGrade}(v) = \mathbf{Aper}_{TGrade}(v) - \mathbf{Aper}_{TGrade}^{baixa}(v)$. No dicionário usado em trabalhos anteriores [24, 25], cada par de imagens era composto por uma imagem em baixa resolução e uma imagem contendo informação em alta frequência. Partindo desta ideia, compomos um dicionário em que cada par é composto pelas imagens $\mathbf{Aper}_{TGrade}^{baixa}(v)$ e $\mathbf{Bord}_{TGrade}(v)$. Da mesma forma como fizemos anteriormente para as imagens de alta frequência, unimos os conjuntos $\{\mathbf{Aper}_{TGrade}^{baixa}(v)\}_n$ em um único conjunto $\{\mathbf{Aper}^{baixa}(\nu)\}^{TViz}$. Temos então um dicionário composto pelo par de conjuntos $\{\mathbf{Aper}^{baixa}(\nu)\}^{TViz}$ e $\{\mathbf{Bord}(\nu)\}^{TViz}$. Para simplificar a leitura, vamos nos referir a esses dois conjuntos como $\{\mathbf{Aper}^{baixa}\}$ e $\{\mathbf{Bord}\}$.

Usando a técnica de ponderação de dicionário de Hung *et al.* [25], como visto na Seção 2.4.7.1, podemos comparar blocos da imagem em baixa resolução a ser super-resolvida \mathbf{Org}^{baixa} com blocos colocalizados das imagens em $\{\mathbf{Aper}^{baixa}\}$. Para cada posição de bloco, temos um vetor de pesos inversamente proporcionais à distorção entre o blocos de \mathbf{Org}^{baixa} e os blocos de $\{\mathbf{Aper}^{baixa}\}$. Esses pesos são usados para compor um bloco da imagem \mathbf{Bord}^{SR} pela ponderação, dos blocos das imagens em $\{\mathbf{Bord}\}$.

Neste teste, a nossa análise leva em conta o resultado da super-resolução em função de: tamanho do dicionário; tamanho do bloco usado na ponderação do dicionário. Novamente, o tamanho do dicionário é representado pelo parâmetro de tamanho da vizinhança $TViz$ da etapa de casamento de gradientes, indexado por v . Relembramos da Seção 4.4 que o tamanho do dicionário é $9 \times TViz$ (três tamanhos de $TGrade$ vezes três conjuntos de imagens compensadas resultantes de duas referências) e, mais uma vez, usamos $1 \leq TViz \leq 15$. Para os tamanhos dos blocos de ponderação, usamos blocos de 2×2 , 4×4 , 8×8 , 16×16 e 32×32 . Testes preliminares mostraram que para blocos de tamanho 64×64 , ou maior, os resultados são consideravelmente inferiores aos demais tamanhos. Os resultados obtidos, referidos como PD-SR, são apresentados para cada sequência na Figura 4.26.

Uma primeira avaliação que pode ser feita desses gráficos é que o uso de blocos de tamanho 16×16 leva aos melhores resultados, exceto para a sequência *news*. Para essa sequência especificamente, a diferença entre o uso do melhor tamanho e o bloco de tamanho 16×16 varia entre $0.01dB$ ($TViz = 6$, quando a curva do Bloco 16×16 atinge seu máximo), passando por $0.06dB$ ($TViz = 8$, quando a curva do Bloco 4×4 atinge seu máximo), até

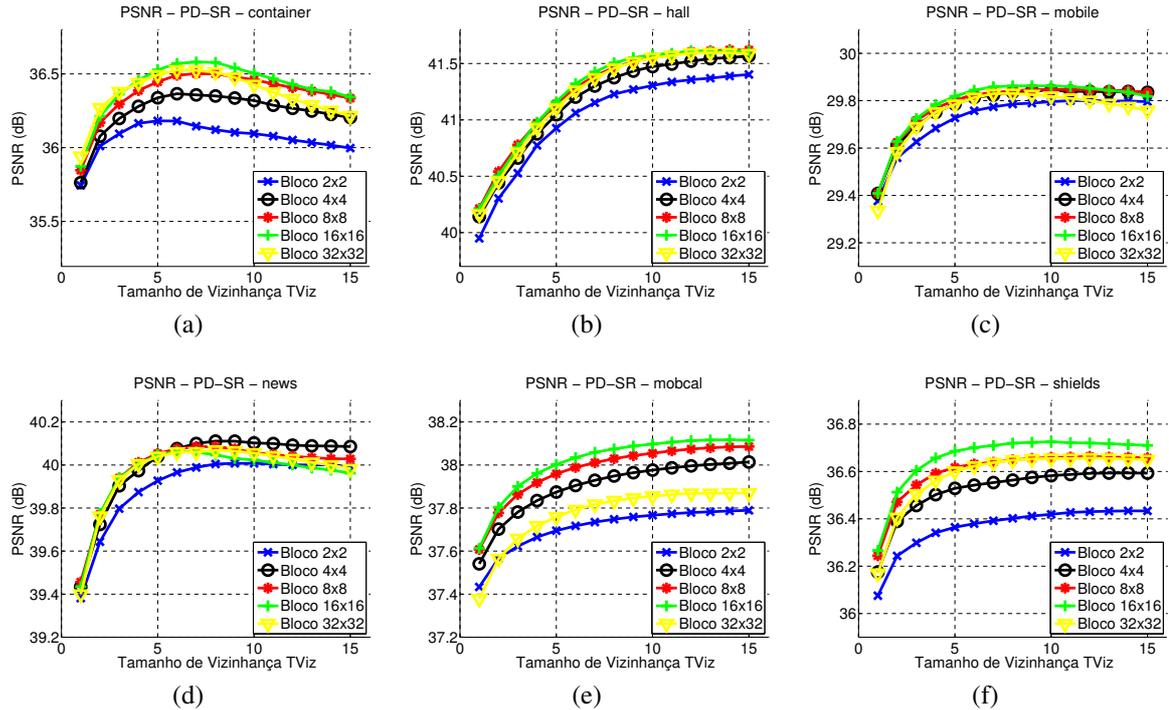


Figura 4.26: Comparação de valores de PSNR resultantes da composição de conjuntos para diferentes tamanhos de dicionário (representados pelo tamanho da vizinhança $TViz$), para todas as seqüências: (a) *container*; (b) *hall*; (c) *mobile*; (d) *news*; (e) *mobcal*; (f) *shields*.

0.12dB ($TViz = 15$, quando a diferença entre as curvas é máxima).

Uma segunda avaliação é que o tamanho do dicionário que leva ao melhor resultado varia de seqüência para seqüência. Além disso, é interessante notar que o uso de dicionários maiores não necessariamente implica resultados melhores. Para todas as seqüências, o valor de PSNR tende a estabilizar. Contudo, para algumas, o valor atinge um máximo e depois estabiliza para um valor menor. Isso indica que, para cada seqüência, existe um tamanho ótimo de dicionário. Essas observações são muito semelhantes àquelas da análise estatística das imagens de alta frequência 4.5. Por isso, buscamos o tamanho de dicionário que, na média, produz os melhores resultados.

Fazemos a mesma análise realizada anteriormente. Normalizamos as curvas de PSNR de cada seqüência referente ao uso do bloco de tamanho 16×16 , e as traçamos juntas, como mostrado na Figura 4.27. A partir dos histogramas dos valores normalizados para cada valor de $TViz$, observamos que, na média, o valor que leva aos melhores resultados é $TViz = 8$, conforme mostrado na Figura 4.28.

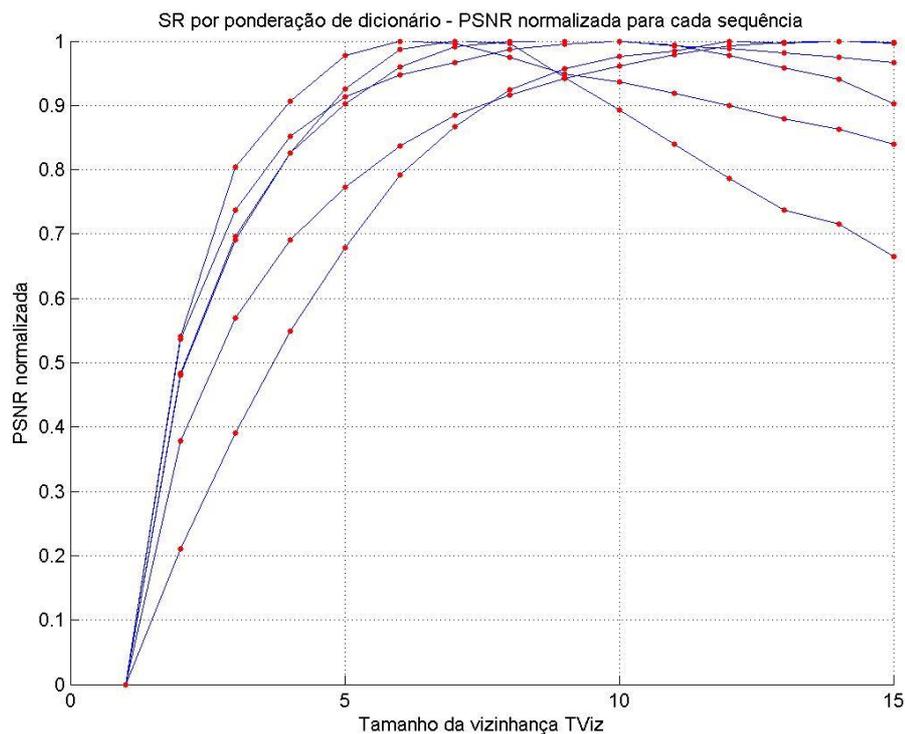


Figura 4.27: Curvas de PSNR dos gráficos mostrados nas Figuras 4.26a a 4.26f para blocos de tamanho 16×16 , normalizadas.

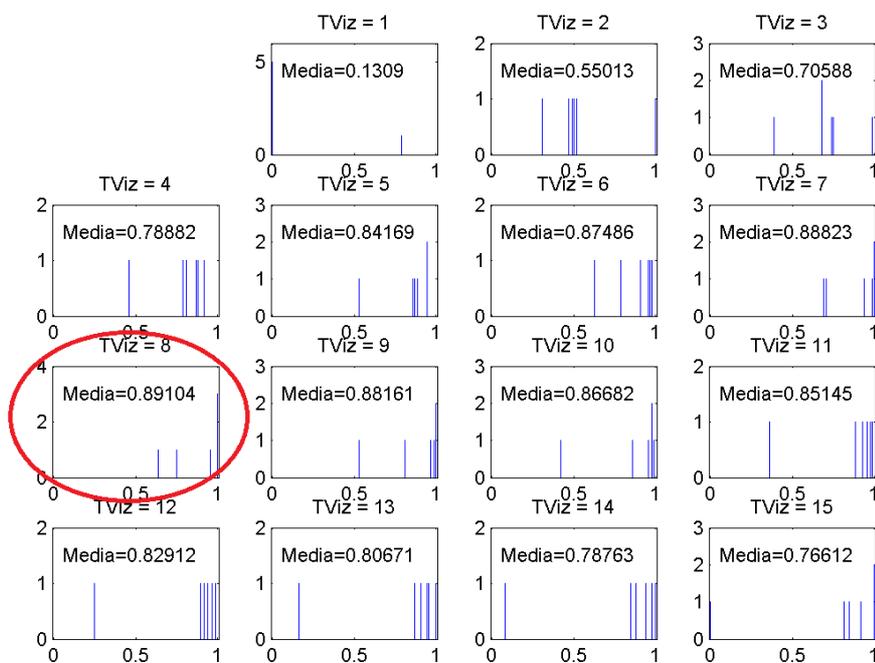


Figura 4.28: Histograma dos valores de PSNR normalizados mostrados na Figura 4.27, para cada valor de $TViz$, referentes à SR por ponderação de dicionário.

4.7 SUPER-RESOLUÇÃO POR PONDERAÇÃO DE DICIONÁRIO COMBINADO

Como este trabalho foi motivado pela busca por superar técnicas de super-resolução baseadas em OBMC, e este último teste usa uma ferramenta proposta para ponderação de dicionário construído usando OBMC, propomos mais uma opção de super-resolução. Compomos um novo dicionário mais completo, resultado da união do dicionário gerado a partir do nosso método com o dicionário gerado usando OBMC, conforme apresentado no Capítulo 2. Como parâmetros para gerar o dicionário, trabalhamos a OBMC com blocos de tamanhos 4×4 , 8×8 , 16×16 , semelhante ao apresentado por Hung *et al.* [25]. Isso produz um dicionário de seis pares de imagens (três tamanhos de blocos vezes duas imagens de referência). Com esse novo dicionário, aplicamos a técnica de ponderação de dicionários, testando novamente os tamanhos de blocos 2×2 , 4×4 , 8×8 , 16×16 e 32×32 . Os resultados obtidos, referidos como PDO-SR, são mostrados na Figura 4.29.

Semelhantemente ao que ocorre com o dicionário usando apenas o nosso método, as curvas têm comportamentos que dependem das sequências. Observamos, porém, que neste caso do dicionário conjunto, o tamanho de bloco que levou ao melhor resultado não foi o mesmo para os dois tamanhos de quadros. Para quadros de tamanho CIF, os blocos de tamanho 8×8 produziram valores mais elevados de PSNR, ao passo que blocos de tamanho 16×16 foram os melhores para quadros de tamanho 720p. Em seguida, buscamos o tamanho do dicionário que leva ao melhor resultado, na média, ainda representado por $TViz$. É importante notar que o tamanho do dicionário conjunto agora é $9 \times TViz + 6$. Mais uma vez, traçamos as curvas de PSNR normalizadas, mostradas na Figura 4.30, com os histogramas mostrados na Figura 4.31.

A Figura 4.30 mostra que para duas sequências, *container* e *hall*, o comportamento é praticamente oposto, com suas curvas de PSNR normalizadas cruzando entre os valores de $TViz = 7$ e $TViz = 8$. Como buscamos o melhor valor, na média, usamos o mesmo critério usado anteriormente. Assim, observando os histogramas na Figura 4.31, afirmamos que o melhor resultado, na média, é obtido com $TViz = 7$.

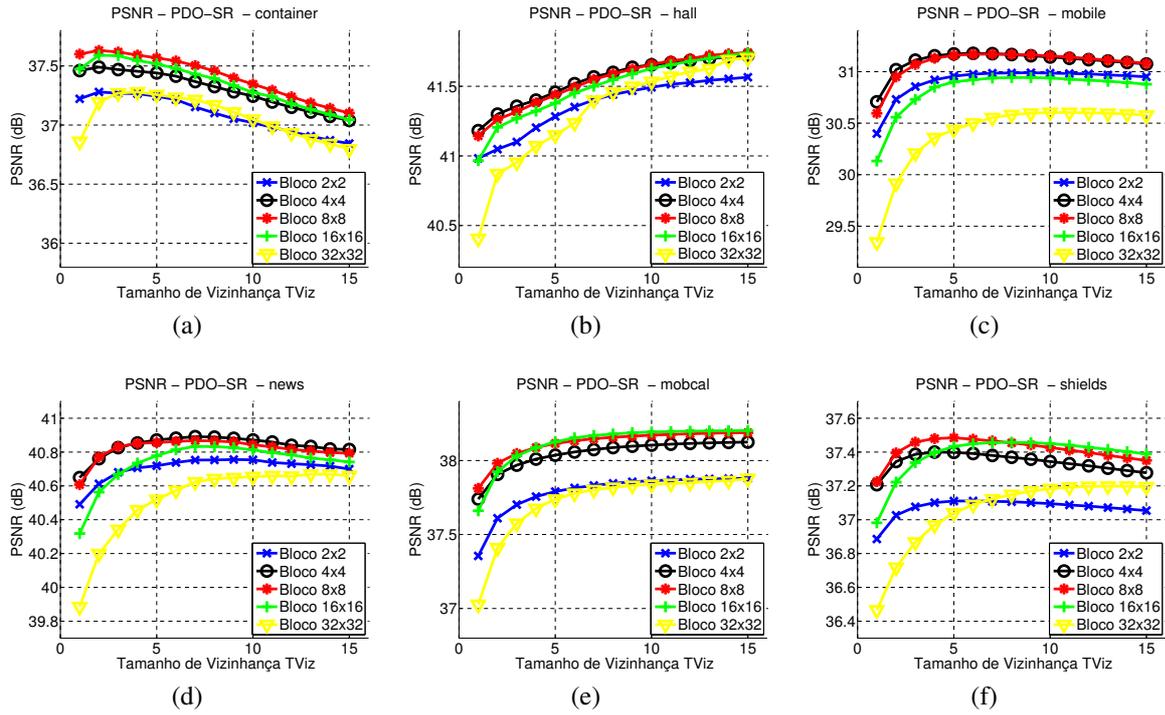


Figura 4.29: Comparação de valores de PSNR resultantes da composição de conjuntos para diferentes tamanos de dicionário (representados pelo tamanho da vizinhança $TViz$), para todas as sequências: (a) *container*; (b) *hall*; (c) *mobile*; (d) *news*; (e) *mobcal*; (f) *shields*.

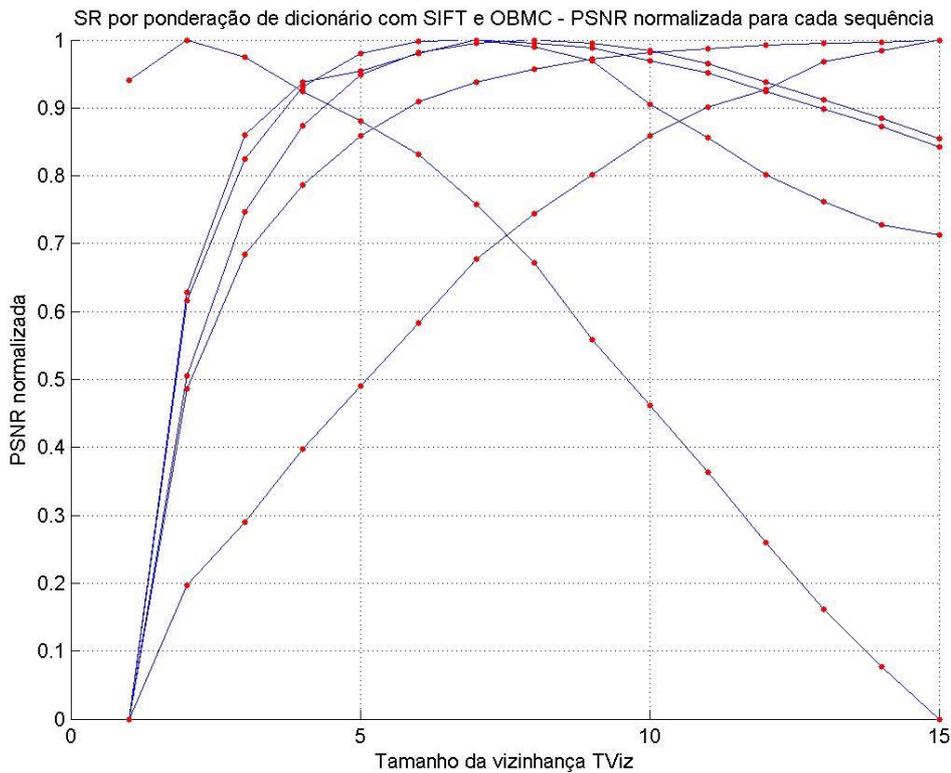


Figura 4.30: Curvas de PSNR dos gráficos mostrados nas Figuras 4.29a a 4.29f normalizadas, para blocos de tamanho 8×8 para sequências CIF e tamanho 16×16 para sequências 720p.

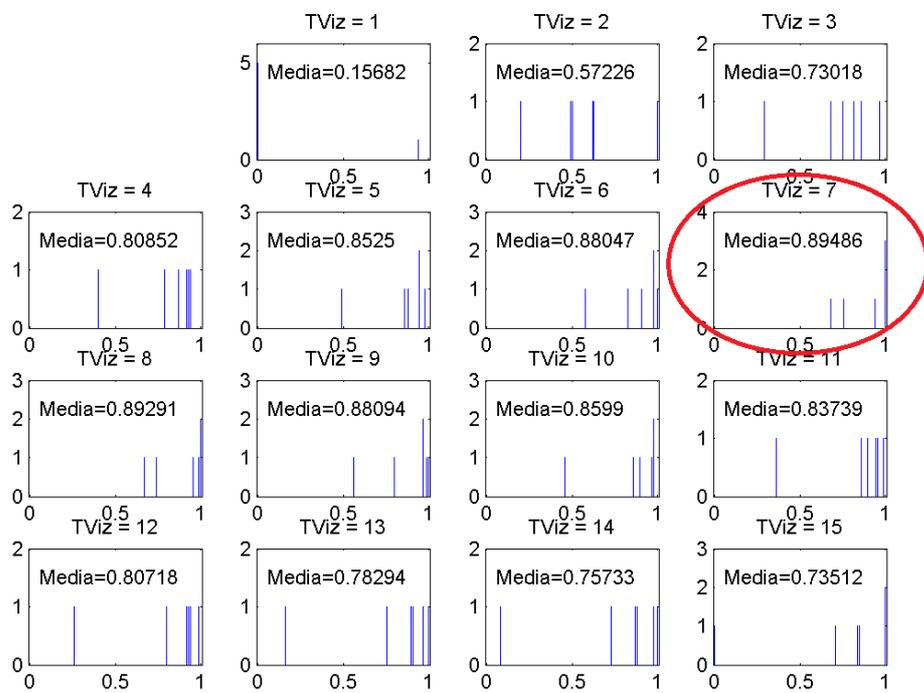


Figura 4.31: Histograma dos valores de PSNR normalizados mostrados na Figura 4.30, para cada valor de $TViz$, referentes à SR por ponderação de dicionário conjunto com OBMC.

4.8 COMPARAÇÃO DE RESULTADOS

Podemos agora comparar os resultados obtidos pelas quatro formas de super-resolução apresentadas. Primeiramente, revisitamos cada forma com os parâmetros que levaram ao melhor resultado, na média.

- Adição direta (AD-SR): $TViz = 5$;
- Análise estatística (AE-SR): $TViz = 9$;
- Ponderação de dicionário (PD-SR): $TViz = 8$, blocos de 16×16 ;
- Ponderação de dicionário com OBMC (PDO-SR): $TViz = 7$, blocos de 8×8 para sequências CIF e blocos de 16×16 para sequências 720p.

Comparamos nossos resultados com aqueles apresentados pelos trabalhos anteriores [24, 25], na Tabela 4.2. Os resultados obtidos por Song *et al.* [24] são indicados por MSR e HSR, ao passo que aqueles obtidos por Hung *et al.* [25] são indicados por DSR (super-resolução por dicionário). Além desses, também comparamos nossos resultados com o estado da arte de super-resolução de imagem única (ISR) proposto por Peleg *et al.* [75]. Mostramos também o valor de PSNR referentes aos quadros interpolados. Quanto aos nossos resultados, para o caso da AD-SR, mostramos o valor de PSNR obtido para o uso de $TGrade$ em que obtivemos o valor mais elevado.

Os valores mostrados em negrito na Tabela 4.2 indicam os melhores valores de PSNR usando apenas o nosso método. Comparando apenas as técnicas AE-SR e PD-SR, a primeira supera a segunda para três das seis sequências testadas, com ganho médio de 0,2 dB, ao passo que a segunda supera a primeira em dois casos, com ganho médio de 0,1 dB. Assim, podemos afirmar que ambas as técnicas produzem resultados muito próximos e são superiores às técnicas apresentadas anteriormente. Porém, a técnica AE-SR é menos dispendiosa computacionalmente, o que pode ser mais vantajoso dependendo da aplicação.

Observando agora a última coluna da Tabela 4.2, vemos que a ponderação de dicionário gerado pelo nosso método conjuntamente com a OBMC traz resultados muito superiores a qualquer um dos demais, tanto dos trabalhos anteriores quanto usando apenas o nosso método, exceto para as sequências *hall* e *mobcal*. No caso específico dessas duas sequências,

Tabela 4.2: Comparação de valores de PSNR para as diferentes técnicas de SR

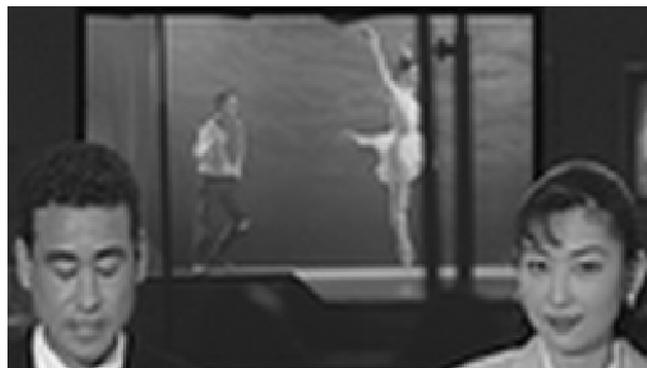
Sequência	Interpolação Lanczos-3	MSR [24]	HSR [24]	DSR [25]	ISR [75]	AD-SR	AE-SR	PD-SR	PDO-SR
<i>container</i>	27,4	31,9	33,2	36,0	29,2	35,8	36,8	36,5	37,4
<i>hall</i>	29,1	37,4	38,0	41,1	30,6	40,9	41,5	41,6	41,6
<i>mobile</i>	22,9	24,5	25,5	27,1	24,1	29,0	29,7	29,9	31,2
<i>news</i>	29,4	31,9	36,1	38,8	32,1	39,2	40,0	40,0	40,9
<i>mobcal</i>	27,7	30,9	31,0	35,0	28,5	37,0	38,2	38,1	38,2
<i>shields</i>	31,1	31,4	32,7	36,0	33,6	36,4	36,9	36,7	37,5

há muito pouco movimento entre o quadro-não-chave e os quadros-chave. Com isso, os dicionários se tornam redundantes. Para as demais sequências, onde há mais movimento, os dicionários se complementam por conta da forma como são gerados, sendo um baseado na semelhança entre *pixels* e outro na semelhança entre características e posteriormente entre gradientes.

As Figuras 4.32 e 4.33 mostram dois exemplos visuais de super-resolução. Em ambas as figuras, mostramos a versão original do quadro-não-chave (previamente à subamostragem), o quadro-não-chave sobreamostrados e duas versões do quadro super-resolvido com as técnicas de adição direta e ponderação de dicionário conjunto com SIFT e OBMC. Nelas podemos observar que o método proposto realmente traz uma sensível melhora na resolução da imagem, comparando com a sobreamostragem. Comparando as duas técnicas, vemos que a AD-SR além de melhorar a resolução, também insere artefatos provenientes do casamento de gradientes com contornos errôneos provocados pela compensação de movimento incorreta. A técnica de PDO-SR, por outro lado, é mais robusta, melhorando a resolução inserindo apenas a informação de alta frequência sem inserção de ruído.



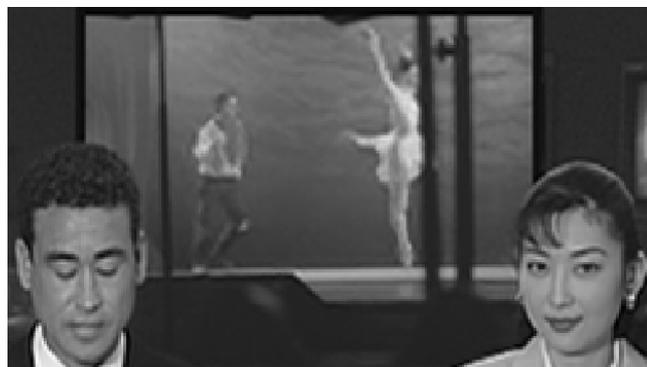
(a)



(b)



(c)



(d)

Figura 4.32: Exemplo comparativo visual para a sequência *news*, com *zoom*: (a) quadro original; (b) quadro interpolado com filtro Lanczos-3; (c) quadro super-resolvido por adição direta; (d) quadro super-resolvido por ponderação de dicionário conjunto com SIFT e OBMC.



(a)



(b)



(c)



(d)

Figura 4.33: Exemplo comparativo visual para a sequência *mobile*, com *zoom*: (a) quadro original; (b) quadro interpolado com filtro Lanczos-3; (c) quadro super-resolvido por adição direta; (d) quadro super-resolvido por ponderação de dicionário conjunto com SIFT e OBMC.

Capítulo 5

Compensação de movimento baseada em agrupamento automático de vetores

O segundo método proposto de solução para o problema apresentado no Capítulo 3, que chamamos de Método de Agrupamento de Vetores, traz duas outras abordagens específicas para as duas etapas de compensação de movimento e casamento de gradientes, distintas daquelas apresentadas no Capítulo 4. Na primeira etapa, a compensação de movimento é feita a partir do agrupamento (*clustering*, em inglês) automático de vetores de correspondência, seguida da definição das regiões de recortes que englobam cada grupo de vetores. Já na segunda etapa, o casamento de gradientes é feito em uma vizinhança V circular que depende das regiões de recorte previamente definidas. Diferentemente daquelas descritas no capítulo anterior, as técnicas aqui apresentadas para a composição dos conjuntos de imagens compensadas $\{Comp(k)\}$ e $\{Comp^{baixa}(k)\}$ e aprimoradas $\{Aper(v)\}$ e $\{Aper^{baixa}(v)\}$ não dependem de parâmetros arbitrários.

5.1 COMPENSAÇÃO DE MOVIMENTO BASEADA EM AGRUPAMENTO DE VETORES

A primeira parte da compensação de movimento, na qual obtemos os conjuntos de imagens compensadas $\{Comp(k)\}$ e $\{Comp^{baixa}(k)\}$, é exatamente a mesma daquela descrita anteriormente. Geramos um fluxo de vetores de correspondência dos descritores SIFT entre as imagens em alta resolução Ref e em baixa resolução sobreamostrada Org^{baixa} . Em seguida, fazemos o agrupamento de vetores de movimento e a definição das regiões de recorte. Ao contrário do método apresentado no Capítulo 4, neste método nós primeiramente definimos o agrupamento dos vetores do fluxo por agrupamento e em seguida, a partir desses agrupamentos, definimos as regiões de recorte. O diagrama da Figura 5.1 mostra a estrutura da compensação de movimento baseada em agrupamento de vetores. Esta etapa é composta dos seguintes passos, com destaque para os passos específicos deste método, que serão detalhados em seguida:

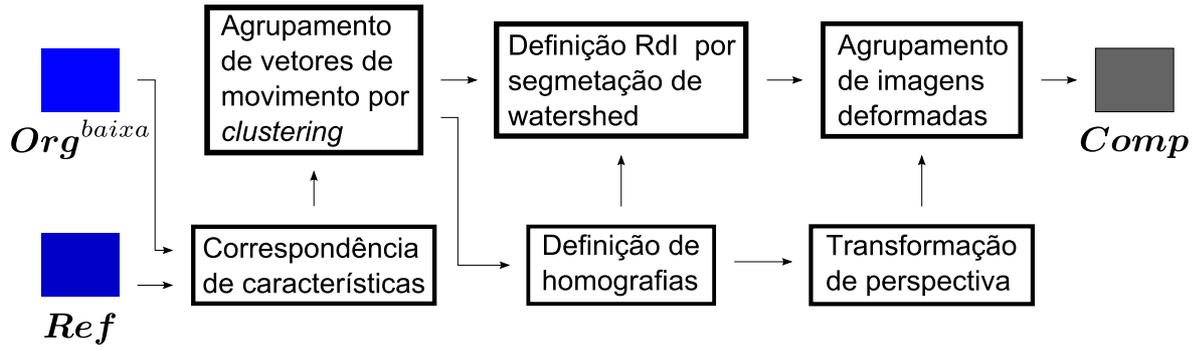


Figura 5.1: Diagrama da compensação de movimento baseada em agrupamento de vetores.

- Detecção de características SIFT nas imagens Org^{baixa} e Ref ;
- Correspondência dos descritores das características detectadas;
- Composição de um fluxo de vetores a partir da diferença de posição dos descritores correspondidos;
- **Agrupamento do fluxo usando técnica de *clustering***;
- **Determinação das regiões de recorte que englobam cada grupo usando transformada de *watershed* [8]**;
- Definição de homografias a partir dos grupos de vetores;
- Transformação de perspectiva da imagem Ref ;
- Composição de um mosaico pelo agrupamento de recortes das imagens transformadas.

5.1.1 Agrupamento de vetores

Uma vez composto o fluxo de vetores de correspondência entre descritores SIFT (conforme descrito na Seção 3.1), realizamos o agrupamento dos vetores. Para realizar o agrupamento, três informações devem ser definidas: a métrica da distância entre os vetores; a métrica da distância entre os grupos de vetores; e a quantidade de grupos. Enquanto as métricas são as mesmas durante todo o processo, o agrupamento será feito para diferentes quantidades de grupos, conforme será explicado adiante.

Para a distância entre vetores, usamos a distância Euclidiana, ou seja, para dois vetores $\mathbf{v}_1 = [x_1, y_1, vx_1, vy_1]^T$ e $\mathbf{v}_2 = [x_2, y_2, vx_2, vy_2]^T$ a distância $d_v(\mathbf{v}_1, \mathbf{v}_2)$ entre eles é:

$$d_v(\mathbf{v}_1, \mathbf{v}_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (vx_1 - vx_2)^2 + (vy_1 - vy_2)^2}, \quad (5.1)$$

São então computadas todas as distâncias entre todos os vetores do fluxo. Isto é feito usando a função *pdist* do MATLAB[®]. Em seguida, para dois grupos r e s , calculamos a distância $d_g(r, s)$ entre eles usando a distância interna quadrática, dada por:

$$d_g(r, s) = \sqrt{\frac{2n_r n_s}{n_r + n_s} \|\bar{r} - \bar{s}\|}, \quad (5.2)$$

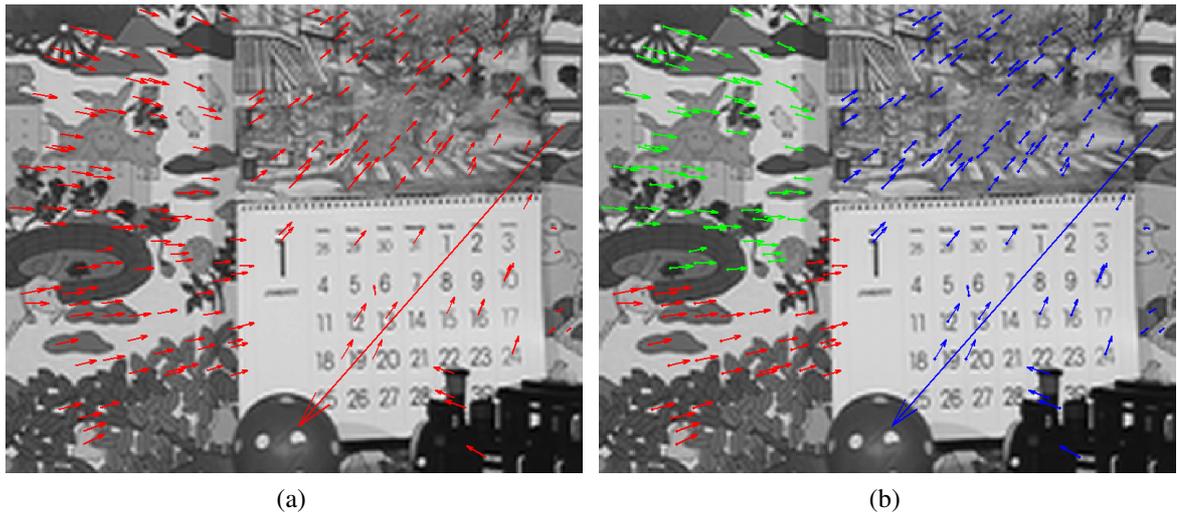


Figura 5.2: Exemplo de fluxo de vetores e seu agrupamento: (a) fluxo não agrupado; (b) fluxo completo separado em três grupos.

em que n_r e n_s são os números de elementos e \bar{r} e \bar{s} são os centroides dos grupos r e s , respectivamente. Com esta distância, realizamos o agrupamento minimizando a variância total dentro de cada grupo, o que é conhecido como método de Ward [76]. Esta métrica é usada para a criação de uma árvore hierárquica de agrupamento usando a função *linkage* do MATLAB[®]. A partir da árvore computada e para um valor k definido de quantidade de grupos, realizamos o agrupamento usando a função *cluster* do MATLAB[®]. A Figura 5.2 mostra o fluxo sobreposto ao quadro-não-chave, com: 5.2a mostrando o fluxo completo não agrupado; 5.2b mostra o fluxo separado em três grupos usando a técnica descrita, cada grupo com uma cor diferente.

5.1.2 Determinação das regiões de recorte

A partir do agrupamento dos vetores, definimos as matrizes de homografias e regiões de recortes referentes a cada grupo. Conforme já foi apresentado no Capítulo 3, as matrizes de homografia são derivadas de pares de pontos usando RANSAC. Lembramos que os vetores são calculados a partir dos pares de coordenadas dos descritores correspondidos e, por isso, vamos tratar de vetores e pares de pontos como a mesma coisa, de forma indiscriminada (um pode ser calculado do outro e vice-versa). As matrizes são derivadas usando novamente a função *findHomography* do OpenCV para cada grupo de vetores. Além da matriz, esta função também retorna quais pares de pontos são ou não discrepantes (do inglês, *outliers*). As Figuras 5.3a e 5.3b mostram o fluxo separado em grupos com os vetores discrepantes em destaque e os vetores discrepantes removidos, respectivamente. É importante ressaltar que os vetores considerados discrepantes são removidos apenas temporariamente e não eliminados em definitivo do fluxo, pois eles podem não ser discrepantes para algum outro agrupamento.

A definição das regiões de recorte, que chamaremos de Regiões de Interesse (RdI), é

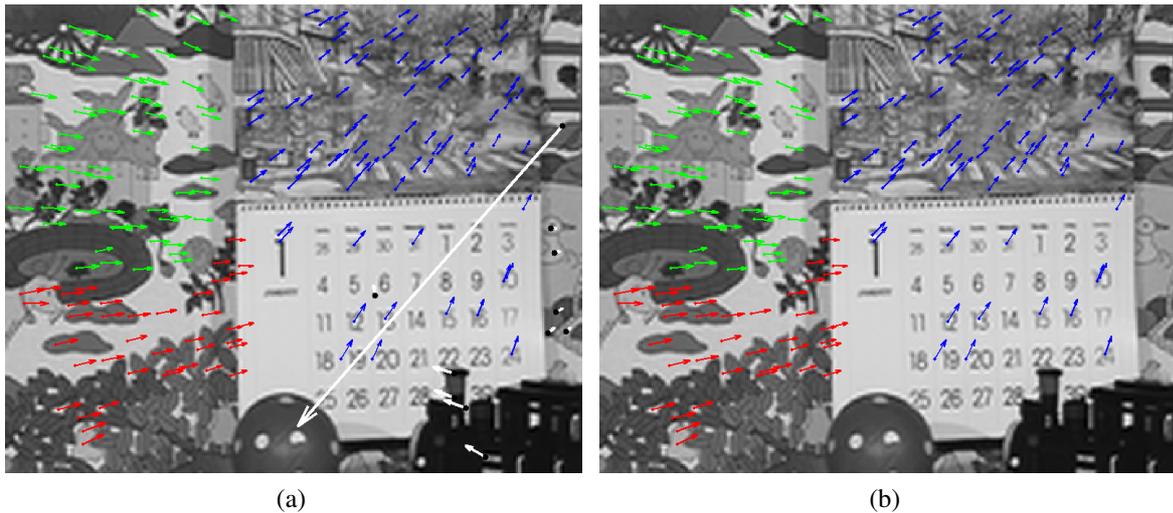


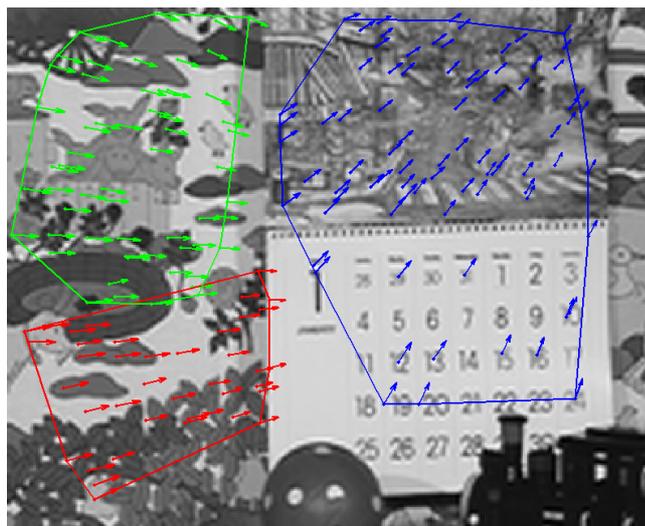
Figura 5.3: Exemplo de detecção e remoção de vetores discrepantes dentro de cada grupo: (a) vetores discrepantes destacados; (b) vetores discrepantes removidos.

feita a partir das coordenadas (x, y) dos vetores não discrepantes de cada grupo. Com isso, apresentamos as etapas executadas para a definição de cada RdI a partir de cada grupo de vetores do fluxo:

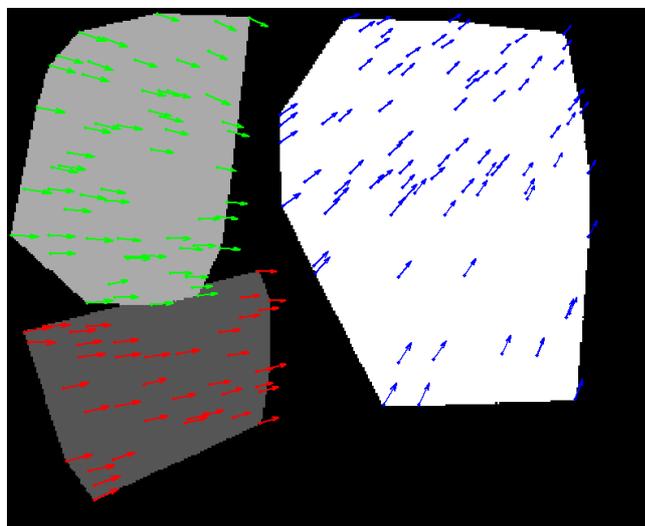
- Remoção dos pontos discrepantes de cada grupo de vetores;
- Cálculo do feixe convexo das coordenadas (x, y) de cada grupo;
- Definição de regiões internas a cada feixe;
- Aplicação da transformada de *watershed* sobre a imagem contendo as regiões dos feixos.

A Figura 5.4 mostra exemplos de cada uma das etapas para encontrar as RdI. Para definir a RdI referente a cada grupo, primeiramente, calculamos o feixe convexo das coordenadas (x, y) dos vetores pertencentes ao grupo, já desconsiderando seus vetores discrepantes. Este feixe determina as bordas de uma região que contém as origens de todos os vetores pertencentes a um grupo. Na Figura 5.4a notamos, porém, que os feixos podem se sobrepor. Isso é corrigido definindo regiões não sobrepostas internas aos feixos convexas, cada uma contendo apenas um grupo de vetores. Essas regiões são chamadas Regiões de Grupos e são usadas para definir as bordas das RdI.

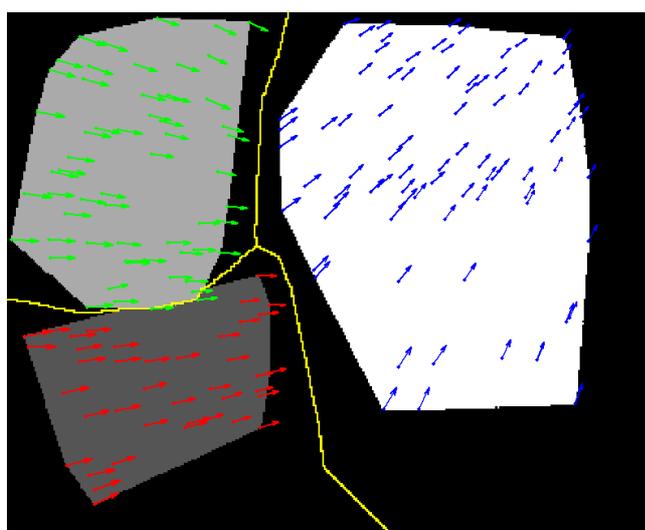
As bordas das RdI são definidas usando a imagem contendo as Regiões de Grupos. Isso é feito gerando uma imagem binária, em que todas as Regiões de Grupo assumem valor 0, enquanto o resto da imagem assume valor 1. Na imagem binária, cada região de valor zero é considerada uma bacia de captação do algoritmo de segmentação por transformada de *watershed*. As linhas de *watershed* definidas pela segmentação são, finalmente, as bordas das RdI, ilustradas na Figura 5.4c.



(a)



(b)



(c)

Figura 5.4: Exemplo de definição das Regiões de Interesse (RdI): (a) feixo convexo; (b) regiões internas aos feixos convexos não-sobrepostas; (c) regiões segmentadas por *watershed*.

5.1.3 Compensação de imagem para diferentes números de grupos de vetores

Assim como foi descrito no Capítulo 3, as imagens compensadas podem ser compostas como um mosaico de recortes. Diferentemente das imagens compensadas pelo método apresentado no Capítulo 4, no método atual as regiões de recorte cobrem toda a área da imagem. Assim, uma vez obtidas as matrizes de homografias e as RdI, podemos compor o par de imagens compensadas $Comp(k)$ e $Comp^{baixa}(k)$, em que o índice k indica o número de grupos em que o fluxo de vetores foi separado, que chamaremos de $NGrupos$. Em outras palavras, o k -ésimo par de imagens $Comp(k)$ e $Comp^{baixa}(k)$ é gerado pela separação do fluxo em $k = NGrupos$ grupos. Relembrando, para cada grupo de vetores g e suas respectivas RdI, com $g \in \{1, 2, \dots, NGrupos\}$, são derivadas:

- uma matriz de homografia que produz uma transformação de perspectiva $\tau_g\{\cdot\}$;
- uma máscara binária M_g gerada por atribuir à RdI valor 1 e valor 0 para o resto da imagem.

Desta forma, para algum valor de $k > 1$, calculamos $Comp(k)$ e $Comp^{baixa}(k)$ por

$$\begin{aligned} Comp(k) &= \sum_{g=1}^k M_g \circ \tau_r\{Ref\}, \\ Comp^{baixa}(k) &= \sum_{g=1}^k M_g \circ \tau_r\{Ref^I\}. \end{aligned} \tag{5.3}$$

em que $Ref^{baixa} = sobre(sub(Ref))$ é novamente a versão reamostrada de Ref e \circ simboliza um produto de Hadamard. A Figura 5.5 mostra as imagens compensadas $Comp(3)$ e $Comp^{baixa}(3)$, com as bordas das RdI sobrepostas.

A composição do par de conjuntos $\{Comp(k)\}$ e $\{Comp^{baixa}(k)\}$ se dá, inicialmente, para o caso específico de $NGrupos = 1$, conforme descrito no Capítulo 3, ou seja, as imagens $Comp(1)$ e $Comp^{baixa}(1)$ são obtidas pela deformação das imagens Ref e Ref^{baixa} , respectivamente, usando apenas uma homografia derivada do fluxo de vetores completo. As demais imagens do par de conjuntos para $NGrupos > 1$ são obtidas pela repetição do processo descrito anteriormente começando com $NGrupos = 2$. A cada passo de agrupamento (para cada valor de $NGrupos$), fazemos algumas verificações iniciais para garantir que todas as etapas sejam executadas corretamente:

- Primeiro, verificamos se todos os grupos têm pelo menos quatro pares de pontos, pois esta quantidade é a mínima necessária para calcular uma matriz de homografia;
- Em seguida, checamos se pelo menos três origens de vetores do grupo são não-colineares para permitir o cálculo do feixe convexo;
- Se algumas das verificações anteriores não forem satisfeitas, os vetores que provocam a falha são definitivamente eliminados do fluxo;



(a)



(b)

Figura 5.5: Exemplo de imagens compensadas com as bordas das RdI sobrepostas: (a) $Comp(3)$; (b) $Comp^{baixa}(3)$.

- Finalmente, em caso de eliminação de vetores, o fluxo é novamente agrupado para o mesmo valor corrente de $NGrupos$.

Após terminada a compensação das imagens para um determinado valor $NGrupos$, seu valor é incrementado em um e o processo é repetido com os vetores remanescentes no fluxo. Com a sucessiva compensação das imagens, chega-se a um ponto em que não é mais possível agrupar o fluxo de modo a satisfazer todas as condições apresentadas acima, ou seja, a cada tentativa de agrupamento, a eliminação de vetores reduz o fluxo de vetores a um ponto em que todos os vetores são eliminados. Neste ponto, encerra-se o processo. Isso garante que os conjuntos $\{Comp(k)\}$ e $\{Comp^{baixa}(k)\}$ têm um tamanho que depende apenas dos vetores do fluxo, sem a necessidade de se arbitrarem parâmetros, como foi feito na técnica apresentada no Capítulo 4. Ao final, unimos as imagens $\{Ref\}$ e $\{Ref^{baixa}\}$ aos conjuntos $\{Comp(k)\}$ e $\{Comp^{baixa}(k)\}$ já compostos para garantirmos um bom casamento de gradientes em caso de cenas de fundo estático.

5.2 CASAMENTO DE GRADIENTES EM VIZINHANÇAS CIRCULARES

Novamente, esta etapa visa à obtenção do par de conjuntos de imagens aprimoradas $\{\mathbf{Aper}(v)\}$ e $\{\mathbf{Aper}^{baixa}(v)\}$ a partir do casamento de gradientes da imagem \mathbf{Org}^{baixa} com aqueles das imagens do conjunto $\{\mathbf{Comp}^{baixa}(k)\}$, para diferentes vizinhanças indexadas por v . Diferentemente da técnica apresentada no capítulo anterior, usamos agora uma vizinhança circular e não mais quadrada por questões de implementação, conforme será detalhado mais à frente. Recapitulando e adaptando para a nova técnica, tomamos $\nabla \mathbf{Org}_{i,j}^{baixa}$ e $\nabla \mathbf{Comp}_{i,j}^{baixa}(k)$ os gradientes dos *pixels* $\mathbf{Org}_{i,j}^{baixa}$ e $\mathbf{Comp}_{i,j}^{baixa}(k)$ pertencentes às imagens \mathbf{Org}^{baixa} e $\{\mathbf{Comp}^{baixa}(k)\}$ na posição (i, j) , respectivamente. Calculamos o índice $\hat{k}_{i,j}^v$ que satisfaz

$$\hat{k}_{i,j}^v = \underset{k}{\operatorname{argmin}} \sum_{s \in V} \sum_{t \in V} \|\nabla \mathbf{Org}_{i+s,j+t}^{baixa} - \nabla \mathbf{Comp}_{i+s,j+t}^{baixa}(k)\|, \quad (5.4)$$

em que V é um círculo de raio v em torno de (i, j) .

Para cada valor de v , calculamos $\mathbf{Aper}(v)$, com *pixels* $\mathbf{Aper}_{i,j}(v) = \mathbf{Comp}_{i,j}(\hat{k}_{i,j}^v)$, e $\mathbf{Aper}^{baixa}(v)$, com *pixels* $\mathbf{Aper}_{i,j}^{baixa}(v) = \mathbf{Comp}_{i,j}^{baixa}(\hat{k}_{i,j}^v)$.

Como buscamos automatizar a tomada de decisão sobre parâmetros, definimos um parâmetro v_{max} que determina o tamanho dos dicionários $\{\mathbf{Aper}(v)\}$ e $\{\mathbf{Aper}^{baixa}(v)\}$, ou seja, $v \in \{1, 2, \dots, v_{max}\}$. O valor de v_{max} é determinado automaticamente, de forma que, para pelo menos uma posição de *pixel* dentro de cada RdI (de todas as imagens no conjunto $\{\mathbf{Comp}^{baixa}(k)\}$), o casamento de gradientes não sofra qualquer influência dos gradientes das bordas da RdI. Para isso, optamos por usar vizinhanças circulares em vez de quadradas. Em outras palavras, buscamos o maior raio de circunferência inscrita a todas as RdI calculadas na etapa de compensação de movimento, para todos os valores de $N\text{Grupos}$. Este processo é feito durante a etapa de compensação de movimento, logo após definição das RdI, usando operações morfológicas com um elemento estruturante octogonal para aproximar o raio da circunferência.

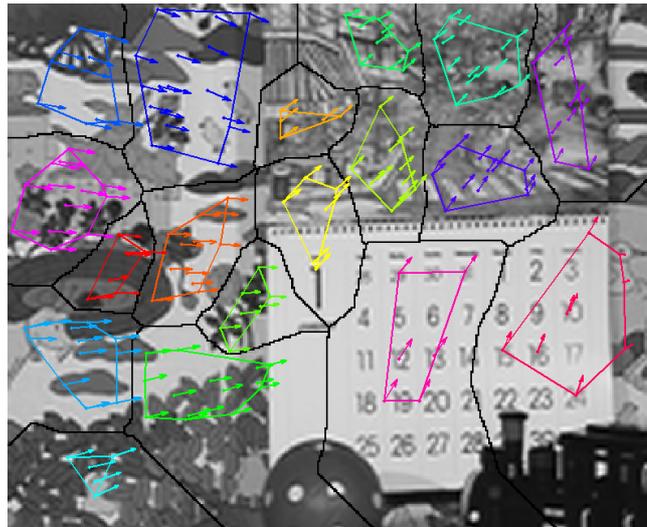
A Figura 5.6 mostra a obtenção da circunferência com $v_{max} = 17$ para o mesmo exemplo da seção anterior, que foi encontrado quando $N\text{Grupos} = 18$. A Figura 5.6a mostra cada grupo de vetores com seus respectivos feixos convexos e as bordas de RdI encontradas. A Figura 5.6b mostra cada RdI com uma cor diferente. Por último, a Figura 5.6c mostra a RdI em que foi encontrada a menor circunferência inscrita.

A Figura 5.6c também mostra, abaixo da circunferência, o octógono usado para a obtenção v_{max} . Esta implementação é feita usando operação morfológica de erosão e dilatação com um elemento estruturante na forma octogonal (que aproxima um elemento estruturante circular). As operações são feitas até que se encontre o maior raio de elemento estruturante que não provoque erosão completa da RdI. O uso de operação morfológica torna a busca

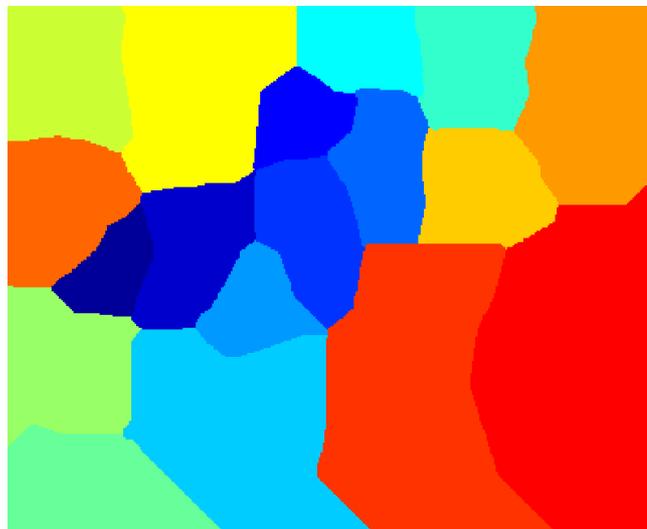
pelo valor de v_{max} menos custosa que o uso de técnicas baseadas na geometria das RdI.

Finalmente, após compostos os conjuntos $\{Aper(v)\}$ e $\{Aper^{baixa}(v)\}$, calculamos as imagens com informação de alta frequência $Bord(v) = Aper(v)$ e $Aper^{baixa}(v)$ a serem usadas para super-resolver a imagem Org^{baixa} . Com este método, testamos duas aplicações distintas. A primeira foi a super-resolução de quadros de vídeos, semelhante àquelas do método anterior, visando à comparação com trabalhos anteriores. A segunda foi com a super-resolução de imagens sob transformações diversas, tanto geométricas quanto não-geométricas, como mudanças de ponto de vista, de escala, de iluminação, etc.

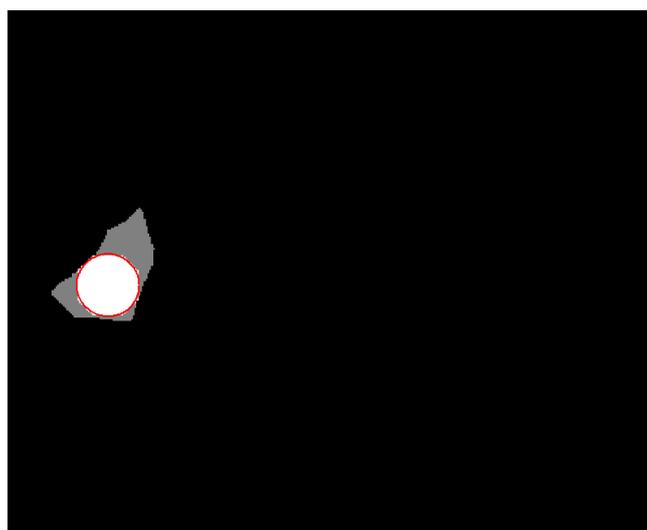
Quanto às técnicas de super-resolução verificadas, testamos a super-resolução por análise estatística (AE-SR) e super-resolução por ponderação de dicionário (PD-SR). Nesta última, testamos as duas composições do dicionário já examinadas, ou seja, um dicionário gerado com as imagens geradas pelo nosso método apenas e um gerado em conjunto com imagens compensadas por OBMC. Nos testes de PD-SR, fizemos uma pequena alteração na composição do dicionário, em relação aos testes executados para o Método das Grades Móveis. Para cada par do dicionário, a imagem com informação de alta frequência continua sendo a imagem $Bord(v)$. Contudo, percebemos que existe uma pequena melhora nos resultados se usarmos como imagem de baixa resolução não mais a imagem $Aper^{baixa}(v)$, mas sim uma nova imagem $A^{baixa}(v) = sobre(sub(Org^{baixa} + Bord(v)))$. Essa melhora se explica por não termos mais bordas de RdI da imagem $Aper^{baixa}(v)$ no cálculo da distorção entre esta e Org^{baixa} para o cálculo dos pesos.



(a)



(b)



(c)

Figura 5.6: Exemplo de obtenção de v_{max} : (a) quadro-não-chave com vetores separados em $N_{Grupos} = 18$, com feixos convexos e bordas de RdIs; (b) todas as RdI; (c) círculo e elemento estruturante octogonal com $v_{max} = 17$.

5.3 SUPER-RESOLUÇÃO DE QUADROS DE VÍDEO

Os primeiros testes que executamos para este segundo método de solução são semelhantes aos que foram feitos para o primeiro método (Método de Grades Móveis), ou seja, foram executados em quadros de vídeos não comprimidos, seguindo as mesmas condições dos trabalhos de Song *et al.* [24] e Hung *et al.* [25]. Como já observamos que o uso de dois quadros é melhor que apenas um, não fazemos esta comparação novamente. Além disso, como o segundo método foi proposto com a ideia de que não houvesse a necessidade de arbitragem de parâmetros, não fizemos nenhum teste neste sentido, como aqueles feitos para determinar o melhor parâmetro $TViz$ nos diferentes testes do método anterior, por exemplo.

5.3.1 Condições de teste

Os testes foram novamente realizados sob condições semelhantes dos trabalhos anteriores de Song *et al.* [24] e Hung *et al.* [25]. Recapitulando:

- A imagem a ser super-resolvida (quadro-não-chave) é uma versão subamostrada do 16º quadro original;
- As imagens usadas como referência (quadros-chave) são o 1º (quadro 0 da sequência) e o 31º (quadro 30 da sequência) quadros originais;
- Foram testadas quatro sequências de tamanho CIF (*Container, Hall, Mobile e News*) e duas sequências de tamanho 720p (*MobCal e Shields*), mostradas;
- A métrica usada para avaliação objetiva dos resultados foi a PSNR.

Além das condições propostas pelo trabalho anterior de Hung *et al.*, testamos nosso método para outro cenário de subamostragem $sub(\cdot)$ apresentado por Peleg *et al.* [75]. Este é um trabalho estado-da-arte de SR com imagem única, ou SI-SR (do inglês *single image super-resolution*). Temos então dois cenários, descritos abaixo:

- Cenário 1: filtragem Lanczos-3 e decimação num fator de escala igual a 2;
- Cenário 2: filtragem bicúbica e decimação num fator de escala igual a 2.

Para todos os cenários, usamos a sobreamostragem $sobre(\cdot)$ em fator de escala igual ao de $sub(\cdot)$, porém o filtro usado é sempre Lanczos-3. Esta escolha foi feita pois este é o filtro que produz os melhores valores de PSNR quando comparando duas imagens X e $sobre(sub(X))$. A Tabela 5.1 mostra os valores médios de PSNR comparando todos os quadros das seis sequências usadas nos testes com suas versões reamostradas para cada combinação de filtros bilinear, bicúbico e Lanczos-3, para um fator de escala igual a 2. A superioridade da sobreamostragem usando o filtro lanczos já era esperada, uma vez que esse filtro (conforme informado no Capítulo 2) é uma aproximação da função de interpolação ideal *sinc*.

Tabela 5.1: Valores de PSNR médios para diferentes combinações de filtros de reamostragem para todos os quadros das sequências: (a) *container*; (b) *hall*; (c) *mobile*; (d) *news*; (e) *mobcal*; (f) *shields*

Filtro de subamostragem	Filtro de sobreamostragem		
	bilinear	bicúbico	Lanczos-3
bilinear	26.5	27.7	28.3
bicúbico	27.7	29.1	29.7
Lanczos-3	28.1	29.5	30.1

(a)

Filtro de subamostragem	Filtro de sobreamostragem		
	bilinear	bicúbico	Lanczos-3
bilinear	25.9	26.7	27.1
bicúbico	26.7	27.6	28.0
Lanczos-3	27.0	27.8	28.3

(b)

Filtro de subamostragem	Filtro de sobreamostragem		
	bilinear	bicúbico	Lanczos-3
bilinear	20.7	21.5	21.9
bicúbico	21.5	22.5	22.9
Lanczos-3	21.9	22.8	23.2

(c)

Filtro de subamostragem	Filtro de sobreamostragem		
	bilinear	bicúbico	Lanczos-3
bilinear	26.5	27.7	28.3
bicúbico	27.7	29.1	29.7
Lanczos-3	28.1	29.5	30.1

(d)

Filtro de subamostragem	Filtro de sobreamostragem		
	bilinear	bicúbico	Lanczos-3
bilinear	27.5	28.4	28.9
bicúbico	28.4	29.5	29.9
Lanczos-3	28.8	29.8	30.2

(e)

Filtro de subamostragem	Filtro de sobreamostragem		
	bilinear	bicúbico	Lanczos-3
bilinear	28.0	29.0	29.5
bicúbico	29.0	30.2	30.7
Lanczos-3	29.5	30.9	31.1

(f)

Tabela 5.2: Comparação de valores de PSNR para diferentes técnicas, sob Cenário 1

Sequência	Interpolação Lanczos-3	ISR [75]	DSR [25]	AE-SR (MGM)	PD-SR (MGM)	PDO-SR (MGM)	AE-SR (MAV)	PD-SR (MAV)	PDO-SR (MAV)
<i>container</i>	27,4	29,1	36,0	36,8	36,5	37,4	36,7	36,7	37,3
<i>hall</i>	29,1	30,6	41,1	41,5	41,6	41,6	41,5	41,9	42,0
<i>mobile</i>	22,9	24,1	27,1	29,7	29,9	31,2	29,3	29,6	30,1
<i>news</i>	29,4	32,0	38,8	40,0	40,0	40,9	39,3	39,6	40,0
<i>mobcal</i>	27,7	28,6	35,0	38,2	38,1	38,2	38,1	38,2	38,2
<i>shields</i>	31,1	33,8	36,0	36,9	36,7	37,5	36,4	36,5	37,2

5.3.2 Resultados experimentais

A super-resolução por análise estatística é praticamente idêntica àquela feita para a solução com grades. A única diferença é que, como comentado, não há necessidade de testes de parâmetros, uma vez que tanto o valor máximo de N_{Grupos} quanto de v_{max} são definidos automaticamente. Assim, novamente compomos uma imagem $Bord^{AE}$ cujos *pixels* $I_{i,j}^{AE}$ são dados pela média aritmética de cada vetor $\hat{i}_{i,j}$. A imagem super-resolvida final é igualmente calculada por $A^{SR} = Org^{baixa} + Bord^{AE}$. Nota-se que para o método MGM, a imagem $Bord^{AE}$ não depende de qualquer parâmetro.

A super-resolução por ponderação de dicionário segue também o que foi descrito no Capítulo 4, exceto pelo que foi comentado anteriormente neste capítulo sobre a alteração das imagens $Aper^{baixa}(v)$ pelas imagens $A^{baixa}(v)$ na composição dos dicionários. Relembrando, cada técnica é apresentada nas tabelas usando as seguintes siglas, em que (MAV) é usado para indicar os resultados usando o método do agrupamento de vetores:

- Análise estatística: AE-SR (MAV);
- Ponderação de dicionário: PD-SR (MAV);
- Ponderação de dicionário com OBMC: PDO-SR (MAV).

Nas Tabelas 5.2 e 5.3, mostramos os resultados comparativos de PSNR para os Cenários 1 e 2. Para todos os Cenários, comparamos nossos resultados com aqueles obtidos pelas técnicas de Hung *et al.* [25] (superior às técnicas de Song *et al.* [24]), indicado por DSR e pela técnica de Peleg *et al.* [75], indicada por ISR. Ambos os códigos foram cedidos gentilmente pelos autores. O código de Peleg *et al.* foi disponibilizado apenas com dicionários prontos, o que não nos permitiu uma comparação com esta solução compondo novos dicionários com as mesmas imagens de referência que usamos. Por questões autorais, o código de composição do dicionário não pôde ser compartilhado, o que poderia ter produzido resultados diferentes. Apenas como referência, também mostramos os valores para a interpolação com filtro Lanczos-3.

Para o Cenário 1 especificamente, a Tabela 5.2 mostra também os valores de PSNR obtidos pelo Método das Grades Móveis, uma vez que foram obtidos no mesmo cenário, indicados por AE-SR (MGM), PD-SR (MGM) e PDO-SR (MGM). Mostramos, também

Tabela 5.3: Comparação de valores de PSNR para diferentes técnicas, sob Cenário 2

Sequência	Interpolação Lanczos-3	ISR [75]	DSR [25]	AE-SR (MAV)	PD-SR (MAV)	PDO-SR (MAV)
<i>container</i>	27,0	29,1	36,4	36,5	36,6	37,1
<i>hall</i>	28,0	30,6	41,1	41,4	41,8	41,9
<i>mobile</i>	22,6	24,1	27,5	28,9	29,3	30,0
<i>news</i>	29,7	32,0	39,4	38,9	39,3	39,8
<i>mobcal</i>	27,5	28,6	34,8	38,1	38,1	38,2
<i>shields</i>	32,7	33,8	36,4	36,5	36,6	37,2

para este cenário, um exemplo comparativo visual para a sequência *mobile* na Figura 5.7, com *zoom*. As figuras mostram o quadro original, bem como os resultados de SR para as técnicas ISR, DSR e PDO-SR (MAV).



(a)



(b)



(c)



(d)

Figura 5.7: Exemplo comparativo visual para a sequência *mobile*, com *zoom*: (a) quadro original; (b) ISR; (c) DSR; (d) PDO-SR (MAV).

5.3.3 Análise dos resultados

A primeira comparação que fazemos diz respeito ao Cenário 1, em que comparamos tanto as diferentes técnicas de SR quanto os nossos dois métodos MGM e MAV. Comparando primeiramente nossa solução com as demais pela Tabela 5.2, podemos afirmar que nosso trabalho é superior aos anteriores, considerando todas as técnicas testadas.

Comparando agora os métodos propostos, bem como as técnicas de AE e PD, não é possível afirmar que existe um método e técnica absolutamente melhor. Para quase todos os testes, o uso da técnica de PDO-SR sob o método MGM se mostrou o mais vantajoso, com ganho médio de 0,5 dB sobre o segundo melhor, ou seja, a técnica PDO-SR sob o método MAV.

Já tínhamos visto no Capítulo 4 que o uso do dicionário conjunto é o que traz os melhores resultados, por explorar a complementariedade da nossa solução com a solução baseada em OBMC. Notamos agora uma superioridade do método MGM sobre o método MAV. Isso se explica pela maior quantidade de informação gerada em ambas as etapas intermediárias no método MGM (tamanhos dos conjuntos de imagens compensadas e imagens aprimoradas), o que leva a uma maior robustez do método. Em outras palavras, quanto mais imagens intermediárias, maior a chance de haver partes com maior semelhança com o quadro-não-chave, tanto no casamento de gradientes quanto na ponderação do dicionário, o que leva a resultados mais verossímeis. Vale notar que, para as sequências com pouco (*hall*) ou muito pouco (*mobcal*) movimento, a técnica PDO-SR (MAV) superou ou se igualou à técnica PDO-SR (MGM), pois o excesso de imagens intermediárias acaba por inserir partes que na verdade atrapalham nas etapas de comparação com o quadro-não-chave.

Para o Cenário 2, observamos novamente a superioridade do método PDO-SR sobre as demais técnicas, tanto anteriores, quanto das técnicas sob o método MAV. Como não foram executados testes para o método MGM sob este cenário, não é possível uma comparação direta. De qualquer maneira, mesmo assumindo que o método MGM seja superior ao método MAV, podemos buscar uma solução de compromisso. O método MAV surgiu como uma solução para o problema de o método MGM depender demasiadamente de arbitragem de parâmetros. Assim, por gerar resultados superiores às demais técnicas de super-resolução, concluímos que o método MAV atingiu o seu objetivo.

O único parâmetro ajustável para esses testes foi o tamanho dos blocos na etapa de ponderação de dicionário. Os tamanhos de blocos que levaram aos melhores resultados, para ambos os cenários, foram 4×4 para sequências CIF e 16×16 para sequências 720p.

5.4 SUPER-RESOLUÇÃO DE IMAGENS SOB TRANSFORMAÇÕES DIVERSAS

A proposta do nosso trabalho é uma solução de super-resolução para quadros de vídeo de resolução mista complementar àquelas baseadas em estimação e compensação de movimento. Neste teste, contudo, buscamos analisar a eficiência da nossa solução para o problema de SR para transformações diversas, comparando-a com o estado da arte. Para isso, usamos um banco de imagens¹ cujos grupos de imagens (cada grupo capturado de uma cena diferente) apresentam as seguintes transformações:

- mudança de ponto de vista;
- mudança de escala;
- rotação;
- mudança de iluminação;
- borramento;
- compressão JPEG.

Cada grupo contém seis imagens. Os nossos testes foram executados em condições semelhantes àquelas dos testes com quadros de vídeo não comprimidos, ou seja, super-resolvemos uma versão subamostrada da 3ª imagem usando a 1ª e a 6ª imagens originais como referência. As Figuras 5.8 a 5.15 mostram as imagens testadas.

Os testes foram realizados sob os Cenários 1 e 2 apresentados na Seção 5.3 e os resultados são apresentados em valores de PSNR.

¹Disponível em <http://www.robots.ox.ac.uk/vgg/research/affine/>

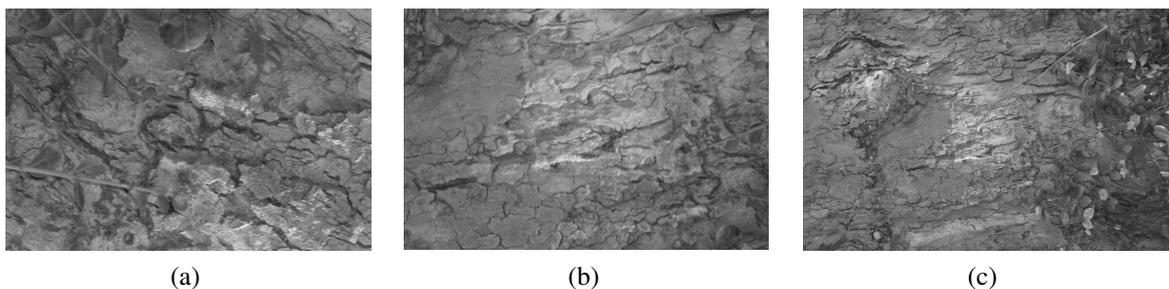


Figura 5.8: Imagens do banco *bark* usadas nos testes: (a) 1ª imagem original, (b) 3ª imagem reescalada (decimado e interpolado) e (c) 6ª imagem original.

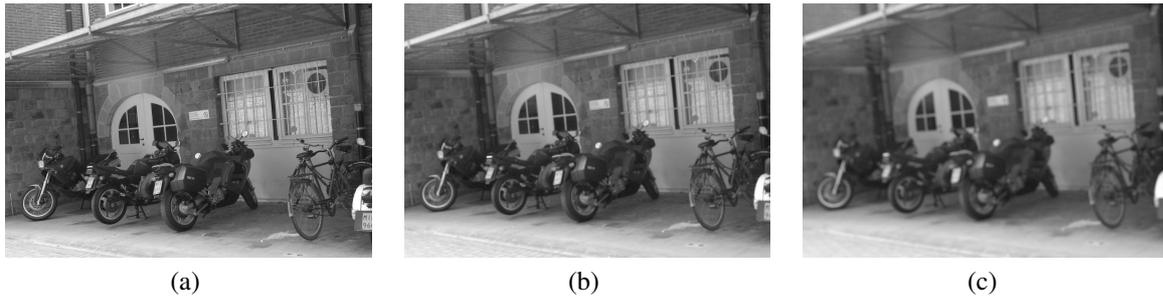


Figura 5.9: Imagens do banco *bikes* usadas nos testes: (a) 1ª imagem original, (b) 3ª imagem reescalada (decimado e interpolado) e (c) 6ª imagem original.

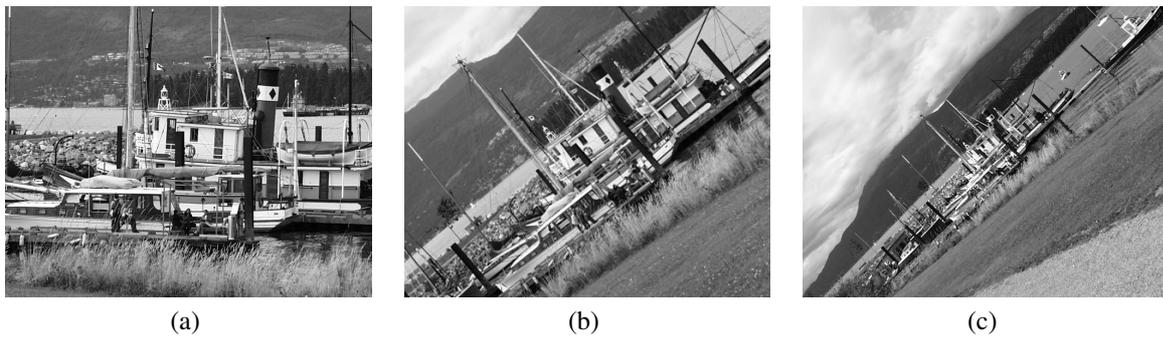


Figura 5.10: Imagens do banco *boat* usadas nos testes: (a) 1ª imagem original, (b) 3ª imagem reescalada (decimado e interpolado) e (c) 6ª imagem original.



Figura 5.11: Imagens do banco *graf* usadas nos testes: (a) 1ª imagem original, (b) 3ª imagem reescalada (decimado e interpolado) e (c) 6ª imagem original.

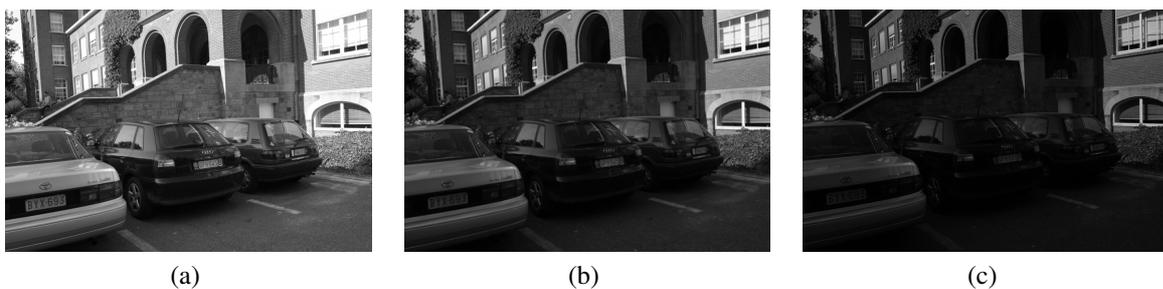


Figura 5.12: Imagens do banco *leuven* usadas nos testes: (a) 1ª imagem original, (b) 3ª imagem reescalada (decimado e interpolado) e (c) 6ª imagem original.

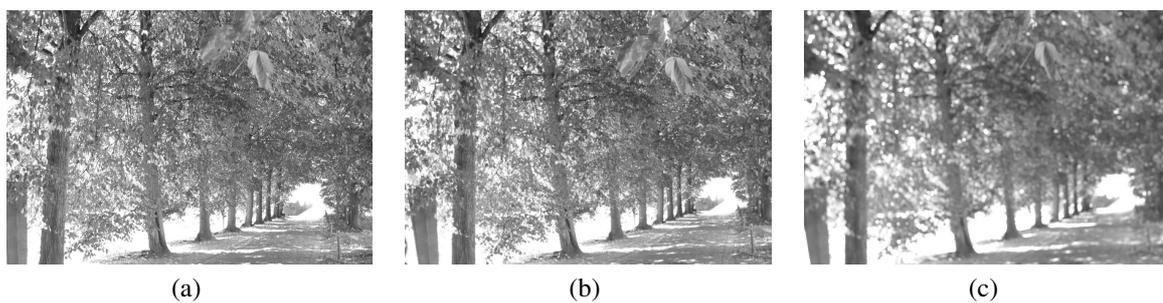


Figura 5.13: Imagens do banco *trees* usadas nos testes: (a) 1ª imagem original, (b) 3ª imagem reescalada (decimado e interpolado) e (c) 6ª imagem original.

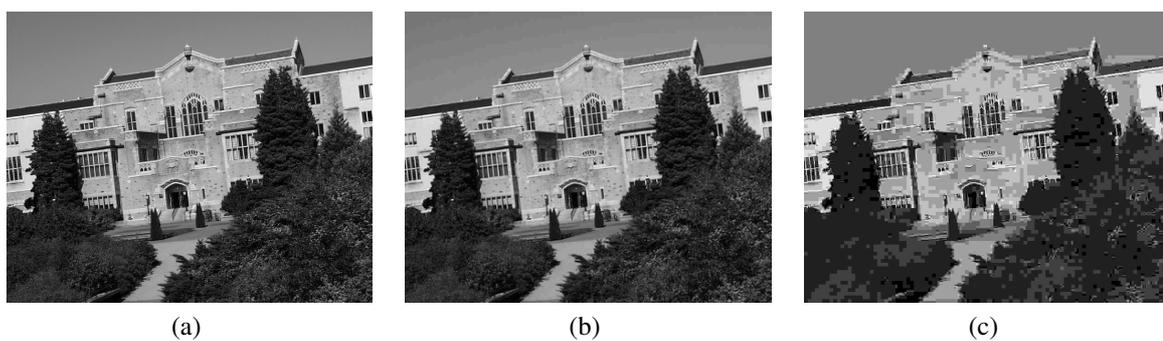


Figura 5.14: Imagens do banco *ubc* usadas nos testes: (a) 1ª imagem original, (b) 3ª imagem reescalada (decimado e interpolado) e (c) 6ª imagem original.

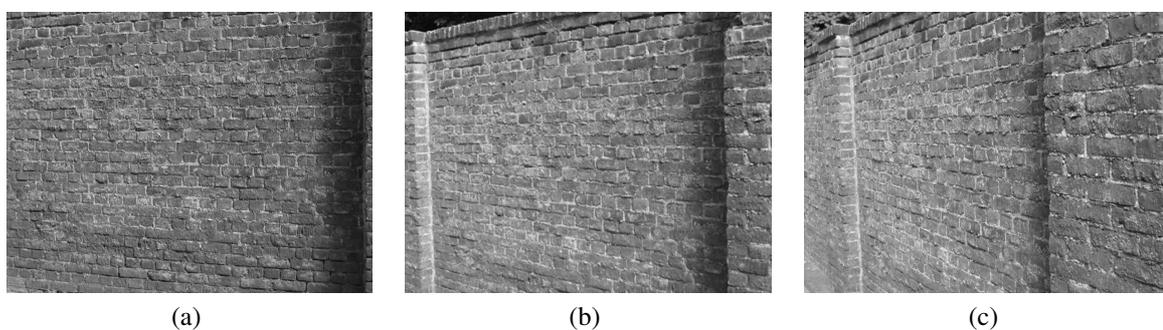


Figura 5.15: Imagens do banco *wall* usadas nos testes: (a) 1ª imagem original, (b) 3ª imagem reescalada (decimado e interpolado) e (c) 6ª imagem original.

5.4.1 Análise dos resultados

A primeira observação que se faz dos resultados mostrados nas Tabelas 5.4 a 5.5 é a superioridade da solução de super-resolução de imagem única de Peleg *et al.* em ambos os cenários, na média. Contudo, podemos fazer uma análise caso a caso.

O banco de imagens *bark* não tem como se beneficiar muito da nossa solução e da solução de Hung *et al.* por conta das partes da cena que cada imagem captura, tendo a solução de Peleg *et al.* ganho médio de aproximadamente 1 dB sobre a técnica PDO-SR (MAV). A primeira imagem em alta resolução representa apenas um pequeno recorte da imagem em baixa resolução, indicando que apenas esta região possa ser super-resolvida. Já a segunda imagem mostra uma captura mais distanciada da cena, o que é representado como uma mudança de escala. A diferença de escalas entre as duas imagens determina o quanto a informação de alta frequência pode ser aproveitada. Se esta mudança é, por exemplo, do mesmo fator de escala do cenário, nada pode ser aproveitado, pois ambas teriam aproximadamente a mesma resolução espacial.

Os bancos de imagens *bikes* e *trees* trazem uma mudança no foco da captura da cena, o que pode ser modelado como uma mudança no quanto a imagem é borrada. Para que a nossa solução tivesse um bom resultado, deveríamos conhecer o filtro de borramento (possivelmente por estimativa). Como isso não faz parte da nossa solução, nosso resultado é inferior, com perda média de 4,1 dB para o banco *bikes* e 0,6 para o banco *trees*.

As imagens do banco *boat* trazem variações de rotação e escala entre as imagens. Esta situação é semelhante ao que ocorre no banco de imagens *bark*, com a diferença de que a primeira imagem em alta resolução do banco *boat* se sobrepõe a uma grande área da imagem em baixa resolução, permitindo a aquisição de informação de alta frequência suficiente para a nossa solução ser superior às demais (no Cenário 1). Com relação ao Cenário 2, o melhor desempenho da solução de Peleg *et al.* pode ser explicado por esta solução ter sido implementada (inclusive os dicionários) especificamente para o uso do filtro bicúbico.

As imagens do banco *graf* trazem uma grande variação na perspectiva de captura da

Tabela 5.4: Comparação de valores de PSNR para diferentes técnicas, sob Cenário 1

Sequência	Interpolação Lanczos-3	ISR [75]	DSR [25]	AE-SR (MAV)	PD-SR (MAV)	PDO-SR (MAV)
<i>bark</i>	33,7	36,8	34,7	35,0	34,9	35,2
<i>bikes</i>	47,3	48,4	46,4	39,9	44,6	45,0
<i>boat</i>	29,6	30,6	29,6	31,0	30,7	30,9
<i>graf</i>	31,6	33,2	32,2	32,8	32,8	33,0
<i>leuven</i>	30,5	31,3	30,9	36,7	36,8	36,7
<i>trees</i>	30,1	31,0	30,7	30,0	30,2	30,3
<i>ubc</i>	28,5	29,5	30,6	30,7	30,2	30,3
<i>wall</i>	29,6	30,2	29,5	33,4	32,9	33,1

Tabela 5.5: Comparação de valores de PSNR para diferentes técnicas, sob Cenário 2

Sequência	Interpolação Lanczos-3	ISR [75]	DSR [25]	AE-SR (MAV)	PD-SR (MAV)	PDO-SR (MAV)
<i>bark</i>	33,6	36,9	34,6	34,8	34,8	35,2
<i>bikes</i>	46,7	48,9	46,0	39,2	43,6	44,1
<i>boat</i>	29,0	31,1	29,3	30,7	30,2	30,5
<i>graf</i>	31,1	33,9	32,0	32,7	32,7	33,0
<i>leuven</i>	30,2	31,4	30,7	36,5	36,5	36,5
<i>trees</i>	29,5	31,2	30,4	29,6	29,6	29,8
<i>ubc</i>	28,1	29,6	30,5	30,7	30,1	30,2
<i>wall</i>	29,2	30,4	29,3	33,2	32,5	32,7

cena. Esta é uma situação em que nossa solução teve desempenho mais próximo à de Peleg *et al.*, com perda de apenas 0,2 dB no Cenário 1. No Cenário 2, a nossa perda foi de 0,9 dB, o que pode ser explicado pelo mesmo que ocorreu para o banco *boat*.

O banco *ubc* traz uma sequência de imagens com diferentes níveis de compressão JPEG. Para este banco, nossa solução tem desempenho muito próximo àquela de Hung *et al.*, com ganho médio de 0,2 dB.

Finalmente, o banco de imagens *wall* é composto por imagens em situação semelhante à do banco *graf*, com transformações de perspectiva, porém não tão pronunciadas. Isso faz com que nossa solução tenha um excelente desempenho, com ganho médio de 3 dB sobre a solução de Peleg *et al.*.

Concluimos que nossa solução tem bom desempenho quando as imagens de referência, além de serem da mesma cena, sejam capazes de fornecer informação de alta frequência para toda a imagem em baixa resolução.

Capítulo 6

Conclusão

Neste trabalho apresentamos uma proposta de solução para o problema de super-resolução baseada em exemplos de uma imagem em baixa resolução e por meio de imagens de alta resolução capturadas de uma mesma cena. A nossa solução é dividida em duas etapas. A primeira etapa consiste na correspondência de descritores de características SIFT, o que gera um fluxo de vetores de movimento. A partir de diferentes agrupamentos desses vetores, calculamos matrizes de homografia para a criação de novas imagens por compensação de movimento usando transformações de perspectiva. Isto leva à composição de um par de conjuntos de imagens compensadas em alta e baixa resolução.

A segunda etapa consiste na composição de um novo par de conjuntos de imagens aprimoradas a partir do casamento de gradientes entre as imagens compensadas em baixa resolução e a imagem em baixa resolução que se deseja super-resolver. Deste par de imagens aprimoradas, calculamos um conjunto de imagens contendo apenas informação de alta frequência, que é usado para calcular uma imagem de alta frequência final a ser adicionada à imagem em baixa resolução.

Nesta linha, propusemos dois métodos distintos de resolver o problema. O primeiro método é baseado, na primeira etapa, na divisão do fluxo de vetor de movimento em grades móveis para a composição das imagens compensadas. Na segunda etapa, geramos as imagens aprimoradas usando o casamento de gradientes em vizinhanças quadradas. No segundo método, usamos o agrupamento dos vetores de movimento do fluxo, seguido de casamento de gradientes em regiões circulares.

6.1 COMPENSAÇÃO DE MOVIMENTO BASEADA EM GRADES MÓVEIS

Para este método, detalhamos a obtenção de informação de alta frequência de quadros-chave para a super-resolução de um quadro-não-chave em vídeos de resolução mista, baseada em grades móveis e casamento de gradientes em vizinhanças quadradas. Em seguida, mostramos quatro técnicas de uso da informação obtida.

Na primeira técnica, adicionamos diretamente ao quadro-não-chave a informação de alta

frequência obtida e buscamos quais parâmetros de tamanho de grade e tamanho de vizinhança levariam ao melhor resultado. Concluimos que o melhor tamanho de vizinhança, na média, é $TViz = 5$, porém os resultados para o tamanho de grade variam muito de sequência para sequência. Propusemos então o uso simultâneo de todas as informações obtidas.

Na segunda técnica, apresentamos o cálculo de uma única imagem contendo informação de alta frequência a partir do valor médio, *pixel a pixel* das imagens de borda obtidas. Verificamos qual a influência do tamanho do conjunto de imagens no resultado final e concluimos que, na média, o Tamanho de Vizinhança $TViz = 9$ (que produz um conjunto de 81 imagens) leva aos melhores resultados.

A terceira técnica consiste na composição de um dicionário em que cada par é composto por uma imagem aprimorada em baixa resolução e uma imagem de alta frequência. Usamos então a técnica de ponderação de dicionário, proposta por Hung *et al.* [25], realizada pela comparação bloco a bloco entre o quadro-não-chave e as imagens em baixa resolução do dicionário. Nossos testes mostraram que, na média, o dicionário gerado com $TViz = 8$ (que contém um total de 72 pares de imagens) e o uso de blocos de tamanho 16×16 produzem os melhores resultados.

Por último, a quarta técnica é semelhante à terceira, porém com a composição de um dicionário com pares de imagens geradas a partir do nosso método e pares de imagens geradas por OBMC. Nossos testes mostraram que, na média, o dicionário gerado com $TViz = 7$ (que contém um total de 69 pares de imagens) e o uso de blocos de tamanho 8×8 para sequências CIF e blocos de 16×16 para sequências 720p produzem os melhores resultados.

Comparando as quatro técnicas, bem como trabalhos anteriores, concluimos que a ponderação de um dicionário composto usando o nosso método e OBMC traz os melhores resultados objetivos.

6.2 COMPENSAÇÃO DE MOVIMENTO BASEADA EM AGRUPAMENTO DE VETORES

Para este método, detalhamos a obtenção de informação de alta frequência de quadros-chave para a super-resolução de um quadro-não-chave em vídeos de resolução mista, baseada em agrupamento de vetores resultantes da correspondência de características SIFT e casamento de gradientes em vizinhanças circulares. Diferentemente do anterior, este método tem a vantagem de não depender de arbitragem de parâmetro, sendo várias decisões tomadas automaticamente pelo algoritmo. Assim, não foi necessário testar o desempenho de parâmetros arbitrados.

Para avaliar este método testamos algumas das mesmas técnicas do método anterior. Além disso, testamos seu desempenho em duas condições bem distintas. Por conta do que

foi observado para o primeiro método, usamos diretamente dois quadros de alta resolução como referência e testamos apenas as técnicas de análise estatística e da ponderação de dicionário, por serem muito superiores e aproveitarem melhor as informações obtidas quando comparado à técnica da adição direta de informação de alta frequência. O método foi comparado com a solução de super-resolução de vídeos de resolução mista de Hung *et al.* [25] e com a solução estado-da-arte de Peleg *et al.* [75] para super-resolução de imagem única usando dicionário pré-concebido.

A primeira condição de teste foi a mesma do método anterior, ou seja, super-resolução de quadros de vídeo não comprimidos. Nesta condição testamos dois cenários de redução de resolução de imagem a ser super-resolvida. Para o primeiro cenário, que consiste na filtragem Lanczos-3 com redução de tamanho por um fator de escala de 2, os testes comparativos mostraram que a automação na decisão de parâmetros levou à redução do desempenho de 0,5 dB, em média, comparado com o método anterior. Com isso, concluímos que o método das grades móveis, para os parâmetros testados, produz imagens nas etapas intermediárias em quantidade maior, trazendo maior robustez à solução proposta. Por outro lado, o método do agrupamento de vetores mantém bom desempenho, com a grande vantagem de automação de decisão de parâmetros. Para o segundo cenário, com filtragem bicúbica e redução do tamanho também por um fator de escala de 2, o método do agrupamento de vetores se mostrou superior às soluções de Hung *et al.* e Peleg *et al.*. Como o método das grades móveis não foi testado sob este cenário, não fizemos uma comparação entre os dois métodos. Com isso, concluímos que nossa solução, e em específico o método do agrupamento de vetores, tem um desempenho superior a outras soluções de super-resolução nas condições testadas.

Na segunda condição de teste, avaliamos o desempenho do método para a super-resolução de imagens capturadas de uma mesma cena, mas sob diferentes transformação. As transformações são tais que o nosso método se mostrou vantajoso apenas para mudanças de iluminação e leve mudanças de perspectiva.

6.3 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Este trabalho alcançou resultados bastante satisfatórios e atingiu o objetivo a que foi proposto. As técnicas propostas se mostraram superiores a outras soluções de super-resolução para a super-resolução de quadros de vídeo em resolução mista e não comprimidos. Mostraram-se superiores, inclusive, ao estado da arte em SR de imagem única para algumas condições específicas. Contudo, ainda há o que ser feito e investigado, tanto para a obtenção de uma técnica mais aprimorada quanto para um melhor entendimento do problema de super-resolução de forma geral. Deixamos então as seguintes propostas de trabalhos futuros.

- Otimizar a implementação da solução para realizar testes exaustivos em mais sequências de vídeo de vários tamanhos de quadro;

- Aplicar a solução proposta para super-resolver vídeos usando imagens estáticas;
- Testar nossa solução com descritores de características binárias, visando à melhoria de eficiência;
- Estudar outras técnicas de agrupamento para separar melhor os vetores do fluxo de acordo com as transformações que proporcionam, visando diminuir a quantidade de passos de agrupamento;
- Estudar de forma mais aprofundada a influência de tamanhos e ponderações de vizinhanças sobre vetores de gradiente;
- Usar segmentação baseada em contornos de objetos em substituição à segmentação de *watershed*, visando calcular transformações de perspectiva mais condizentes com os movimentos reais dos objetos na cena;
- Usar filtragem adaptativa para estimar o filtro usado no processo de subamostragem da imagem em baixa resolução a ser super-resolvida, o que dispensa a necessidade de teste com cenários específicos;
- Analisar a possibilidade de processamento paralelo (possivelmente com GPU) para acelerar a execução do algoritmo;
- Analisar qual o limite de diminuição de escala para fatores não inteiros para a solução proposta.

REFERÊNCIAS BIBLIOGRÁFICAS

- 1 YANG, Jianchao; HUANG, Thomas. Super-resolution imaging. In: _____. [S.l.]: CRC Press, 2010. cap. Image super-resolution: Historical overview and future challenges, p. 3–35.
- 2 HUANG, T. S.; TSAI, R. Y. Multiframe image restoration and registration. *Adv. Comput. Vis. Image Process.*, v. 1, n. 2, p. 317–339, 1984.
- 3 NASROLLAHI, K.; MOESLUND, T. B. Super-resolution: a comprehensive survey. *Machine Vision and Applications*, v. 25, n. 6, p. 1423–1468, Agosto 2014.
- 4 PARK, S.C.; PARK, M.K.; KANG, M.G. Super-resolution image reconstruction: a technical overview poly-Si TFT. *IEEE Signal Processing Mag.*, v. 20, n. 3, p. 21–36, May 2003.
- 5 FREEMAN, W.T.; JONES, T.R.; PASZTOR, E.C. Example-based super-resolution. *IEEE Comput. Graph. Appl.*, v. 22, p. 56–65, March 2002.
- 6 FERREIRA, R.U.; HUNG, E.M.; QUEIROZ, R.L. de. Video super-resolution based on local invariant features matching. In: *Proc. IEEE Intl. Conf. on Image Processing (ICIP'12)*. Orlando, USA: [s.n.], 2012. p. 877–880.
- 7 FERREIRA, R.U.; HUNG, E.M.; QUEIROZ, R.L. de. Clustering of matched features and gradient matching for mixed-resolution video super-resolution. In: *Proc. IEEE Intl Symposium on Circuits and Systems (ISCAS'15)*,. [S.l.: s.n.], 2015. p. 1202–1205.
- 8 GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. 3rd edition. ed. NJ, USA: Prentice-Hall, 2006.
- 9 KOSCHAN, Andreas; ABIDI, Mongi. *Digital Color Image Processing*. [S.l.]: John Wiley & Sons, Inc., 2008.
- 10 RICHARDSON, Iain E. G. *H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia*. New York, NY, USA: John Wiley & Sons, Inc., 2003.
- 11 Canada Centre for Mapping and Earth Observation. *Fundamentals of Remote Sensing*. 2014. Disponível em: http://www.nrcan.gc.ca/sites/www.nrcan.gc.ca/files/earthsciences/pdf/resource/tutor/fundam/pdf/fundamentals_e.pdf.
- 12 OPPENHEIM, Alan V.; SHAFER, Ronald W.; BUCK, John R. *Discrete-time Signal Processing*. [S.l.]: Prentice Hall, 1998.
- 13 KEYS, Robert G. Cubic convolution interpolation for digital image processing. *IEEE Trans. Acoustics, Speech, and Signal Processing*, v. 29, n. 6, p. 1153–1160, 1981.
- 14 BURGER, W.; BURGE, M.J. *Principles of Digital Image Processing: Core Algorithms*. [S.l.]: Springer, 2009.

- 15 DINIZ, Paulo Sergio R.; SILVA, Eduardo A. B. da; NETTO, Sergio L. *Processamento Digital de Sinais - Projeto e Analise de Sistemas*. [S.l.]: Bookman, 2014.
- 16 CLARK, J.J.; PALMER, M.R.; LAURENCE, P.D. A transformation method for the reconstruction of functions from nonuniformly spaced samples. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-33, p. 1151–1165, 1985.
- 17 TOM, B.C.; KATSAGGELOS, A.K. Reconstruction of a high-resolution image by simultaneous registration, restoration, and interpolation of low-resolution images. In: *Proc. 1995 IEEE Int. Conf. Image Processing*. [S.l.: s.n.], 1995.
- 18 STARK, H.; OSKOUI, P. High resolution image recovery from image-plane arrays, using convex projections. *J. Opt. Soc. Am. A*, v. 6, p. 1715–1726, 1989.
- 19 FARSIU, Sina; FARSIU, Sina; FARSIU, Sina; ROBINSON, Dirk; ROBINSON, Dirk; ELAD, Michael; ELAD, Michael; MILANFAR, Peyman; MILANFAR, Peyman. Advances and challenges in super-resolution. *International Journal of Imaging Systems and Technology*, v. 14, p. 47–57, 2004.
- 20 KIM, S.P.; BOSE, N.K.; VALENZUELA, H.M. Recursive reconstruction of high resolution image from noisy undersampled multiframes. *IEEE Trans. Acoust., Speech, Signal Processing*, v. 38, p. 1013–1027, 1990.
- 21 SU, W.; KIM, S. P. High-resolution restoration of dynamic image sequences. *International Journal of Imaging Systems and Technology*, v. 5, n. 4, p. 330–339, 1994.
- 22 RHEE, S.H.; KANG, M.G. Discrete cosine transform based regularized high-resolution image reconstruction algorithm. *Opt. Eng.*, v. 38, n. 8, p. 1348–1356, 1999.
- 23 LIU, Ce; SUN, Deqing. On bayesian adaptive video super resolution. *IEEE Trans. Pattern Anal. Mach. Intell.*, v. 36, n. 2, p. 346–360, Feb 2014.
- 24 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 Syst. Video Technol.*, v. 21, n. 3, p. 274–285, March 2011.
- 25 HUNG, E.M.; QUEIROZ, R.L. de; BRANDI, F.; OLIVEIRA, K.F.; MUKHERJEE, D. Video super-resolution using codebooks derived from key frames. *IEEE Trans. Circuits Syst. Video Technol.*, v. 22, n. 9, p. 1321–1331, Sep. 2012.
- 26 LOWE, D.G. Distinctive image features from scale-invariant keypoints. *Intl. Journal of Computer Vision*, v. 60, n. 2, p. 91–110, Jan 2004.
- 27 GOPRO. *Hero 4 Black - Manual do Usuário*. Disponível em: <http://pt.gopro.com/support/product-manuals-support>.
- 28 MUKHERJEE, D.; MACCHIAVELO, B.; QUEIROZ, R.L. de. A simple reversed-complexity Wyner-Ziv video coding mode based on a spatial reduction framework. In: *Proc. IST/SPIE Symp. on Electronic Imaging, Visual Communications and Image Processing*. San Jose, USA: [s.n.], 2007.
- 29 NOGAKI, S.; OHTA, M. An overlapped block motion compensation for high quality motion picture coding. In: *Proc. IEEE Int. Symp. Circuits Systems*. [S.l.: s.n.], 1992. v. 1, p. 184 – 187.

-
- 30 ITU-R. *Recommendation ITU-R BT.500-13: Methodology for the subjective assessment of the quality of television pictures*. [S.l.], 2012. Disponível em: <http://www.itu.int/rec/R-REC-BT.500-13-201201-I/en>.
 - 31 BJONTEGAARD, G. *Calculation of average PSNR differences between RD-curves*. [S.l.], 2001.
 - 32 TUYTELAARS, Tinne; MIKOLAJCZYK, Krystian. Local invariant feature detectors: A survey. *FnT Comp. Graphics and Vision*, v. 3, n. 3, p. 177–280, 2008.
 - 33 CANNY, John. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1986.
 - 34 HARRIS, Chris; STEPHENS, Mike. A combined corner and edge detector. In: *In Proc. of Fourth Alvey Vision Conference*. [S.l.: s.n.], 1988. p. 147–151.
 - 35 SMITH, Stephen M.; BRADY, J. Michael. Susan – a new approach to low level image processing. *Int. J. Comput. Vision*, Kluwer Academic Publishers, Hingham, MA, USA, v. 23, n. 1, p. 45–78, maio 1997. ISSN 0920-5691. Disponível em: <http://dx.doi.org/10.1023/A:1007963824710>.
 - 36 ROSTEN, Edward; DRUMMOND, Tom. Machine learning for high-speed corner detection. In: *Proceedings of the 9th European Conference on Computer Vision - Volume Part I*. Berlin, Heidelberg: Springer-Verlag, 2006. (ECCV'06), p. 430–443. ISBN 3-540-33832-2, 978-3-540-33832-1. Disponível em: http://dx.doi.org/10.1007/11744023_34.
 - 37 BEAUDET, P. R. Rotationally invariant image operators. In: *Proceedings of the 4th International Joint Conference on Pattern Recognition*. Kyoto, Japan: [s.n.], 1978. p. 579–583.
 - 38 LINDBERG, Tony. Feature detection with automatic scale selection. *International Journal of Computer Vision*, v. 30, p. 79–116, 1998.
 - 39 WITKIN, Andrew P. Scale-space filtering. In: *Proceedings of the Eighth International Joint Conference on Artificial Intelligence - Volume 2*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1983. (IJCAI'83), p. 1019–1022. Disponível em: <http://dl.acm.org/citation.cfm?id=1623516.1623607>.
 - 40 KOENDERINK, Jan J. The structure of images. *Biological cybernetics*, v. 50, n. 5, p. 363–370, 1984.
 - 41 LINDBERG, Tony. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, p. 224–270, 1994.
 - 42 MIKOLAJCZYK, K.; SCHMID, C. Indexing based on scale invariant interest points. In: *Proceedings of the 2001 International Conference on Computer Vision*. [S.l.: s.n.], 2001. p. 525–531.
 - 43 MIKOLAJCZYK, K.; SCHMID, C. An affine invariant interest point detector. In: *Proceedings of the 7th European Conference on Computer Vision-Part I*. London, UK, UK: Springer-Verlag, 2002. (ECCV '02), p. 128–142. ISBN 3-540-43745-2. Disponível em: <http://dl.acm.org/citation.cfm?id=645315.649184>.

-
- 44 MIKOLAJCZYK, Krystian; SCHMID, Cordelia. Scale & affine invariant interest point detectors. *Int. J. Comput. Vision*, Kluwer Academic Publishers, Hingham, MA, USA, v. 60, n. 1, p. 63–86, out. 2004. ISSN 0920-5691. Disponível em: <http://dx.doi.org/10.1023/B:VISI.0000027790.02288.f2>.
- 45 HEALEY, Glenn; SLATER, David. Using illumination invariant descriptors for recognition. In: *Conference on Computer Vision and Pattern Recognition, CVPR 1994, 21-23 June, 1994, Seattle, WA, USA*. [s.n.], 1994. p. 355–360. Disponível em: <http://dx.doi.org/10.1109/CVPR.1994.323851>.
- 46 FLORACK, Luc; ROMENY, Bart M. ter Haar; KOENDERINK, Jan J.; VIERGEVER, Max A. General intensity transformations and differential invariants. *Journal of Mathematical Imaging and Vision*, v. 4, n. 2, p. 171–187, 1994. Disponível em: <http://dblp.uni-trier.de/db/journals/jmiv/jmiv4.html#FlorackRKV94>.
- 47 MINDRU, Florica; TUYTELAARS, Tinne; GOOL, Luc Van; MOONS, Theo. Moment invariants for recognition under changing viewpoint and illumination. *Comput. Vis. Image Underst.*, Elsevier Science Inc., New York, NY, USA, v. 94, n. 1-3, p. 3–27, abr. 2004. ISSN 1077-3142. Disponível em: <http://dx.doi.org/10.1016/j.cviu.2003.10.011>.
- 48 BAUMBERG, Adam. Reliable feature matching across widely separated views. In: *CVPR*. IEEE Computer Society, 2000. p. 1774–1781. ISBN 0-7695-0662-3. Disponível em: <http://dblp.uni-trier.de/db/conf/cvpr/cvpr2000.html#Baumberg00>.
- 49 SCHAFFALITZKY, Frederik; ZISSERMAN, Andrew. Multi-view matching for unordered image sets, or “how do i organize my holiday snaps?”. In: *Proceedings of the 7th European Conference on Computer Vision-Part I*. London, UK, UK: Springer-Verlag, 2002. (ECCV ’02), p. 414–431. ISBN 3-540-43745-2. Disponível em: <http://dl.acm.org/citation.cfm?id=645315.649164>.
- 50 FREEMAN, William T.; ADELSON, Edward H. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 13, p. 891–906, 1991.
- 51 CARNEIRO, Gustavo; JEPSON, Allan D. Multi-scale phase-based local features. In: *CVPR (1)*. [S.l.]: IEEE Computer Society, 2003. p. 736–743.
- 52 LOWE, D.G. Object recognition from local scale-invariant features. In: *Proc. IEEE Intl. Conf. on Computer Vision (ICCV’99)*. Corfu, Greece: [s.n.], 1999. p. 1150–1157.
- 53 MIKOLAJCZYK, K.; SCHMID, C. A performance evaluation of local descriptors. *IEEE Trans. Pattern Analysis and Machine Intelligence*, v. 27, n. 10, p. 1615–1630, October 2005.
- 54 KE, Yan; SUKTHANKAR, Rahul. Pca-sift: A more distinctive representation for local image descriptors. In: *Proceedings of the 2004 IEEE Conference on Computer Vision and Pattern Recognition*. Washington, DC, USA: IEEE Computer Society, 2004. (CVPR’04), p. 506–513. Disponível em: <http://dl.acm.org/citation.cfm?id=1896300.1896374>.

-
- 55 BAY, H.; TUYTELAARS, T.; GOOL, L. Van. Surf: Speeded up robust features. In: *Proceedings of the European Conference on Computer Vision*. [S.l.: s.n.], 2006.
- 56 TUYTELAARS, Tinne; SCHMID, Cordelia. Vector quantizing feature space with a regular lattice. In: *Proceedings of the 2007 International Conference on Computer Vision*. [S.l.: s.n.], 2007.
- 57 CALONDER, Michael; LEPETIT, Vincent; STRECHA, Christoph; FUA, Pascal. Brief: Binary robust independent elementary features. In: *Proceedings of the 11th European Conference on Computer Vision: Part IV*. Berlin, Heidelberg: Springer-Verlag, 2010. (ECCV'10), p. 778–792. ISBN 3-642-15560-X, 978-3-642-15560-4. Disponível em: <http://dl.acm.org/citation.cfm?id=1888089.1888148>.
- 58 RUBLEE, Ethan; RABAUD, Vincent; KONOLIGE, Kurt; BRADSKI, Gary. Orb: An efficient alternative to sift or surf. In: *Proceedings of the 2011 International Conference on Computer Vision*. Washington, DC, USA: IEEE Computer Society, 2011. (ICCV '11), p. 2564–2571. ISBN 978-1-4577-1101-5. Disponível em: <http://dx.doi.org/10.1109/ICCV.2011.6126544>.
- 59 LEUTENEGGER, Stefan; CHLI, Margarita; SIEGWART, Roland Y. Brisk: Binary robust invariant scalable keypoints. In: *Proceedings of the 2011 International Conference on Computer Vision*. Washington, DC, USA: IEEE Computer Society, 2011. (ICCV '11), p. 2548–2555. ISBN 978-1-4577-1101-5. Disponível em: <http://dx.doi.org/10.1109/ICCV.2011.6126542>.
- 60 ALAHI, Alexandre; ORTIZ, Raphael; VANDERGHEYNST, Pierre. Freak: Fast retina keypoint. In: *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Washington, DC, USA: IEEE Computer Society, 2012. (CVPR '12), p. 510–517. ISBN 978-1-4673-1226-4. Disponível em: <http://dl.acm.org/citation.cfm?id=2354409.2354903>.
- 61 WU, Song; LEW, Michael S. Riff: Retina-inspired invariant fast feature descriptor. In: *Proceedings of the ACM International Conference on Multimedia*. New York, NY, USA: ACM, 2014. (MM '14), p. 1129–1132. ISBN 978-1-4503-3063-3. Disponível em: <http://doi.acm.org/10.1145/2647868.2654994>.
- 62 HEINLY, Jared; DUNN, Enrique; FRAHM, Jan-Michael. Comparative evaluation of binary features. In: *Proceedings of the 12th European Conference on Computer Vision - Volume Part II*. Berlin, Heidelberg: Springer-Verlag, 2012. (ECCV'12), p. 759–773. ISBN 978-3-642-33708-6. Disponível em: http://dx.doi.org/10.1007/978-3-642-33709-3_54.
- 63 MAIR, Elmar; HAGER, Gregory D.; BURSCHKA, Darius; SUPPA, Michael; HIRZINGER, Gerhard. Adaptive and generic corner detection based on the accelerated segment test. In: *Proceedings of the 11th European Conference on Computer Vision: Part II*. Berlin, Heidelberg: Springer-Verlag, 2010. (ECCV'10), p. 183–196. ISBN 3-642-15551-0, 978-3-642-15551-2. Disponível em: <http://dl.acm.org/citation.cfm?id=1888028.1888043>.

-
- 64 JUAN, Luo; GWON, Oubong. A Comparison of SIFT, PCA-SIFT and SURF. *International Journal of Image Processing (IJIP)*, v. 3, n. 4, p. 143–152, 2009. Disponível em: <http://www.cscjournals.org/csc/manuscript/Journals/IJIP/volume3/Issue4/IJIP-51.pdf>.
- 65 HARTLEY, R. I.; ZISSERMAN, A. *Multiple View Geometry in Computer Vision*. Second. [S.l.]: Cambridge University Press, ISBN: 0521540518, 2004.
- 66 FISCHLER, Martin A.; BOLLES, Robert C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, ACM, New York, NY, USA, v. 24, n. 6, p. 381–395, jun. 1981. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/358669.358692>.
- 67 YUAN, Z.; YAN, P.; LI, S. Super resolution based on scale invariant feature transform. In: *Proc. IEEE Intl. Conf. on Audio, Language and Image Processing (ICALIP'08)*. Shanghai, China: [s.n.], 2008. p. 1550–1554.
- 68 AMINTOOSI, M.; FATHY, M.; MOZAYANI, N. Regional varying image super-resolution. In: *Proc. Intl. Joint Conf. on Computational Sciences and Optimization (CSO'09)*. Sanya, Hainan, China: [s.n.], 2009. p. 913–917.
- 69 NEMRA, A.; AOUF, N. Robust invariant automatic image mosaicing and super resolution for uav mapping. In: *Proc. IEEE Intl. Symp. on Mechatronics and its Applications (ISMA'09)*. Sharjah, UAE: [s.n.], 2009. p. 1–7.
- 70 HSU, C.-C.; LIN, C.-W. Image super-resolution via feature-based affine transform. In: *Proc. IEEE Intl. Workshop on Multimedia Signal Processing (MMSP'11)*. Hangzhou, China: [s.n.], 2011. p. 1–5.
- 71 YUE, H.; YANG, J.; SUN, X.; WU, F. SIFT-based image super-resolution. In: *Proc. IEEE Intl. Conf. on Circuits and Systems (ISCAS'13)*. Beijing, China: [s.n.], 2013. p. 2896 – 2899.
- 72 MATLAB. *version 8.1.0.604 (R2013a)*. Natick, Massachusetts: The MathWorks Inc., 2013.
- 73 BRADSKI, G. *Dr. Dobb's Journal of Software Tools*, 2000.
- 74 HESS, R. An open source SIFT library. In: *Proc. ACM Multimedia (ACMM10)*. Firenze, Italy: [s.n.], 2010.
- 75 PELEG, Tomer; ELAD, Michael. A statistical prediction model based on sparse representations for single image super-resolution. *IEEE Transactions on Image Processing*, v. 23, n. 6, p. 2569–2582, 2014.
- 76 WARD JR., J. H. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, v. 58, p. 236–244, 1963.