TÓPICOS EM RECONSTRUÇÃO DE IMAGENS E VÍDEOS

Daniel Rodrigues Pipa

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Elétrica.

Orientador: Eduardo Antônio Barros da Silva

Rio de Janeiro
Novembro de 2012

TÓPICOS EM RECONSTRUÇÃO DE IMAGENS E VÍDEOS

Daniel Rodrigues Pipa

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof. Eduardo Antônio Barros da Silva, Ph.D.

Prof. Luiz Wagner Pereira Biscainho, D.Sc.

Prof. Paulo Sergio Ramirez Diniz, Ph.D.

Prof. Hae Yong Kim, D.Sc.

Prof. Marcelo Victor Wüst Zibetti, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
NOVEMBRO DE 2012

Pipa, Daniel Rodrigues

Tópicos em Reconstrução de Imagens e Vídeos/Daniel Rodrigues Pipa. – Rio de Janeiro: UFRJ/COPPE, 2012.

XVII, 219 p.: il.; 29, 7cm.

Orientador: Eduardo Antônio Barros da Silva

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2012.

Referências Bibliográficas: p. 193 – 219.

1. Deconvolução. 2. Super-resolução. 3. Total variation. 4. Interpolação. 5. Problemas inversos. I. Silva, Eduardo Antônio Barros da. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

A Talita, minha esposa.

Agradecimentos

Agradeço ao Centro de Pesquisa e Desenvolvimento da Petrobras (CENPES) pelo incentivo e permissão para realização do doutorado.

Agradeço ao Professor Eduardo, pela orientação, incentivo, paciência e trocas de ideias inspiradoras ao longo desses 6 anos que englobaram meu mestrado e doutorado.

Agradeço ao Professor Truong Nguyen, que me recebeu em 2011 na Universidade da Califórnia, San Diego (UCSD), durante o ano que realizei o doutorado sanduíche.

Agradeço a Stanley Chan, com quem trabalhei na UCSD e tive ricas conversas e discussões que geraram trabalhos e despertaram entendimentos mais profundos sobre o tema desta tese.

Agradeço aos autores do *template* CoppeTeX v2.2, que preparam essa maneira efetiva e eficaz de gerar teses nos padrões exigidos pela UFRJ/COPPE.

Agradeço a minha família (mãe, pai e irmãs) que sempre me apoiaram mesmo nas empreitadas mais difíceis e sempre incentivaram a busca de novos conhecimentos.

Agradeço, finalmente, a minha esposa Talita, que sempre esteve do meu lado, apoiando, incentivando, entendendo, escutando, não importando a cidade nem o país, e sem quem eu certamente não teria conseguido chegar aqui.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

TÓPICOS EM RECONSTRUÇÃO DE IMAGENS E VÍDEOS

Daniel Rodrigues Pipa

Novembro/2012

Orientador: Eduardo Antônio Barros da Silva

Programa: Engenharia Elétrica

Durante a aquisição de imagens e vídeos, ocorrem com frequência degradações que comprometem a qualidade da imagem. As degradações mais comuns são: borramento causado pela resposta óptica do sistema ou *point spread function* (PSF); redução da resolução ou resolução abaixo do desejado para a cena; ruído aditivo; e ruído de padrão fixo (comum em sistemas de imageamento infravermelho). Em geral, essas degrações causam perda de informação a princípio irreversível.

Este trabalho apresenta um estudo sobre algumas técnicas modernas de recuperação de imagens e vídeo que sofreram tais degradações, além de propor extensões. Para suprir a falta de informação, são exploradas características de imagens naturais através de modelos (p. ex. deconvolução por *total variation* (TV), interpolação direcional, modelos autorregressivos) e observações adicionais da cena obtidas pelos diversos quadros consecutivos de um vídeo (p. ex. super-resolução e remoção de ruído de padrão fixo).

Os métodos propostos geram resultados encorajadores, indicando que a linha de pesquisa seguida tem grande potencial. Finalmente, reavaliando-se a literatura mais atual, novas direções e trabalhos futuros são delineados, inspirados em descobertas recentes na área de problemas inversos em processamento de imagens.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

TOPICS IN IMAGE AND VIDEO RECONSTRUCTION

Daniel Rodrigues Pipa

November/2012

Advisor: Eduardo Antônio Barros da Silva

Department: Electrical Engineering

During the acquisition of images and videos, degradation often corrupts image quality. The most common types of degradation are: blurring, caused by the optics of the system; reduction of resolution or resolution below desired quality; additive noise and fixed pattern noise, the latter being very common in infrared imaging systems.

This work presents a study and extensions of some modern techniques for reconstruction of images and videos which have undergone such types of degradation. Since degradation often implies in loss of information, clues are obtained by exploiting natural image characteristics through image models and by considering consecutive frames of a video as different views of the same scene.

The proposed methods show encouraging results, attesting the potential of the approaches used. Finally, future works and new directions are drawn based on recent works and new discoveries related to inverse problems in image processing.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Acrônimos

- AMI : *Autoregressive model interpolation* - Interpolação por modelos autorregressivos
- AMID : *Autoregressive model interpolation and deblurring* - Interpolação e *deblurring* por modelos autorregressivos
- AP : *Affine projection* [*methods*]
- AWGN : *Additive white Gaussian noise* - Ruído branco Gaussiano aditivo
- BM3D : *Block-matching 3D transform denoising*
- BTV : *Bilateral total variation*
- DCT : *Discrete cosine transform* - Transformada discreta de cossenos
- DFT : *Discrete Fourier transform* - Transformada discreta de Fourier
- FFT : *Fast Fourier transform* - Transformada rápida de Fourier
- FoE : *Field of Experts*
- FPN : *Fixed-pattern noise* - Ruído de padrão fixo
- HR : *High-resolution* [*image*] - [Imagem em] alta resolução
- ICA : *Independent component analysis* - Análise de componentes independentes
- IRFPA : *Infrared focal plane array*
- IID : *Independent and identically distributed* [*random vector*] - [Vetor aleatório com componentes] - independentes e identicamente distribuídas
- ISA : *Independent subspace analysis* - Análise de subespaços independentes
- IR : *Infrared* [*video*] - [Vídeo na] faixa do infravermelho
- LMS : *Least mean square* [*estimation*]
- LR : *Low-resolution* [*image*] - [Imagem em] baixa resolução
- MAP : *Maximum a posteriori* [*estimator*]
- ML : *Maximum likelihood* [*estimator*]
- MRF : *Markov random fields* - Campos aleatórios de Markov
- NUC : *Nonuniformity corretion* [*algorithms*]
- PAR : *Piecewise autoregressive* [*models*] - [Modelos] autorregressivos por partes
- PCA : *Principal components analysis* - Análise de componentes principais
- PDF : *Probability density function* - Função densidade de probabilidade
- PLE : *Piecewise linear estimator* - Estimador linear por partes
- PSF : *Point spread function* - Função de espalhamento de ponto

- PSNR : *Peak signal-to-noise ratio* - Relação sinal-ruído de pico
- RLS : *Recursive least squarse* - Mínimos quadrados recursivos
- SAI : *Soft-decision autoregressive interpolation*
- SVD : *Singular value decomposition* - Decomposição por valores singulares
- SSIM : *Structural similarity* - Similaridade Estrutural
- TV : *Total variation*

Resumo da Notação

- Escalares: a, b, λ, μ.
- Vetores: $\mathbf{f}$, $\mathbf{g}$.
- Matrizes: $\mathbf{W}$, $\mathbf{H}$.
- Coordenadas contínuas: x, y ou $\mathbf{x} = [x_1,\, x_2]^{\mathrm{T}}$.
- Coordenadas discretas: n_1, n_2 ou $\mathbf{n} = [n_1,\, n_2]^{\mathrm{T}}$.
- Funções de variáveis contínuas: $f(x, y)$ ou $f(\mathbf{x})$.
- Funções de variáveis discretas: $f(n_1, n_2)$ ou $f(\mathbf{n})$.
- Vetores representam imagens em sua versão com colunas empilhadas.
- Transformações lineares são representadas por matrizes.
- *Patches* (blocos) de uma imagem $\{\mathbf{f}_{(i)}\}_{1 \leq i \leq I}$ ou $\{\mathbf{f}_{(i)}\}_{i \in \mathcal{I}}$
- Quadros de um vídeo: $\{\mathbf{f}_k\}_{1 \leq k \leq K}$.
- Função custo: $J(\mathbf{f})$.
- Atualizações em algoritmos recursivos: $\mathbf{f}_{q+1} = \mathbf{f}_q - \mu \nabla_{\mathbf{f}} J(\mathbf{f})$.

Capítulo 1

Introdução

Sistemas de imageamento, como qualquer aquisição de sinais, introduzem distorções e ruídos no sinal adquirido. Quando essas degradações são imperceptíveis, em geral não há nenhum problema[1]. Porém, quando a imagem foi comprometida e por qualquer motivo soluções triviais[2] para a melhoraria de sua qualidade foram descartadas, pode-se recorrer a técnicas de processamento de imagem para se estimar uma imagem hipotética adquirida por um sistema de aquisição perfeito. Em sistemas de imageamento *digital*, especialmente, destacam-se como degradações típicas: ruído aditivo, borramento (*blurring*) e redução de resolução ou resolução abaixo do esperado para a cena.

Uma primeira simplificação que comumente se adota, e que também será adotada neste trabalho, é considerar apenas problemas lineares [1]. Dois motivos justificam tal simplificação: problemas lineares, ou eventualmente problemas não-lineares que foram linearizados, surgem de modelos lineares que são considerados suficientes para a maioria das aplicações. Além disso, existem diversos algoritmos eficientes e bem-estabelecidos para solução problemas lineares, fruto de décadas de estudos e desenvolvimentos na área. Felizmente, as degradações típicas listadas acima podem ser razoavelmente representadas por modelos lineares.

Usando o ferramental de álgebra linear, pode-se modelar a aquisição de uma imagem por

$$\mathbf{g} = \mathbf{W}\mathbf{f} + \boldsymbol{\nu}, \tag{1.1}$$

onde

- $\mathbf{f}$ representa uma imagem hipotética (original, sem degradação, indisponível e que se deseja estimar). $\mathbf{f}$ é vetor coluna em que cada elemento representa a

[1] Mesmo que degradações não comprometam a qualidade de imagem em um primeiro momento, elas podem se tornar visíveis se for necessário algum processamento posterior, p. ex. melhoramento de contraste pode acentuar ruído.

[2] Efetuar nova aquisição de imagem ou vídeo, substituir e/ou corrigir o sensor/camera por um superior, mais moderno, etc.

Degradação	Modelo	Efeito sobre a imagem
Ruído aditivo	$\mathbf{g} = \mathbf{f} + \boldsymbol{\nu}$	Flutuações aleatória nos *pixels*
Ruído de padrão fixo	$\mathbf{g} = \mathbf{Af} + \mathbf{b}$	Flutuações fixas nos *pixels*
Borramento *blurring*	$\mathbf{g} = \mathbf{Hf}$	Atenuação e possível eliminação de certas frequências (componentes)
Subamostragem	$\mathbf{g} = \mathbf{Rf}$	*Aliasing*

Tabela 1.1: Degradações típicas em sistemas de imageamento digital

intensidade luminosa de um *pixel* da imagem em sua forma digital. Usualmente, as colunas da imagem são "empilhadas" para se formar o vetor.

- $\mathbf{g}$ é um vetor, usando a mesma lógica de formação de $\mathbf{f}$, que representa a imagem adquirida e potencialmente degradada.
- $\mathbf{W}$ é uma matriz, portanto uma operação linear em $\mathbf{f}$ ou ainda um operador convolucional, que modela degradações na imagem que podem ser combinadas, p. ex. $\mathbf{W} = \mathbf{RHM}$, onde
  - $\mathbf{M}$ representa possíveis movimentos na imagem, como rotações, translações, etc.
  - $\mathbf{H}$ representa a operação de borramento (*blurring*), que nada mais é que redução do conteúdo em frequência. Tal efeito é normalmente causado pela resposta óptica do sistema, chamada de *point spread function* (PSD).
  - $\mathbf{R}$ representa uma operação de subamostragem, ou seja, alguns *pixels* são eliminados em um *grid* regular.
- $\boldsymbol{\nu}$ representa ruído aditivo que pode contaminar a imagem durante a aquisição. Esse ruído pode ser proveniente da eletrônica de aquisição, perturbações atmosféricas, entre outros.

A Tabela 1.1 resume as degradações típicas, os modelos observação e seus efeitos na imagem final.

Geralmente, essas degradações causam perda de informação irreversível. Por exemplo no caso do ruído aditivo, sua natureza aleatória impede sua descrição determinística e, portanto, dada uma aquisição que gere $\mathbf{g}$, a operação de remoção de ruído

$$\hat{\mathbf{f}} = \mathbf{g} - \boldsymbol{\nu} \tag{1.2}$$

não é possível.

Um outro tipo de ruído comum em sistemas de imageamento na faixa do infravermelho é o ruído de padrão fixo ou *fixed-pattern noise* (FPN) com observação descrita por

$$\mathbf{g} = \mathbf{A}\mathbf{f} + \mathbf{b}, \tag{1.3}$$

onde $\mathbf{A}$ é uma matriz diagonal que representa um ruído multiplicativo, ou seja, um ganho não desejado e individual em cada *pixel* e $\mathbf{b}$ representa um desvio ou *off-set* também individual e não desejado.

Diferentemente do ruído aleatório, no ruído de padrão fixo os parâmetros ganho $\mathbf{A}$ e desvio $\mathbf{b}$ são constantes ou variam lentamente, podendo ser aproximados por constantes. Logo, a imagem poderia ser recuperada por

$$\hat{\mathbf{f}} = \mathbf{A}^{-1}(\mathbf{g} - \mathbf{b}). \tag{1.4}$$

Contudo, normalmente não se conhecem $\mathbf{A}$ e $\mathbf{b}$ e seus valores devem ser estimados a partir das observações degradadas que, além disso, incluem com frequência ruído aleatório.

No caso de borramento (*blurring*), pode ocorrer atenuação e mesmo eliminação de certas componentes de frequência, o que torna impossível sua recuperação. Finalmente, se a imagem for subamostrada pode ocorrer *aliasing*, processo sabidamente irreversível.

Embora fundamentalmente irreversíveis, existem diversas técnicas para reverter tais processos. Em geral, recorre-se a outras fontes para suprir e auxiliar na recuperação da informação que foi perdida. A informação extra pode estar, por exemplo, em um conhecimento prévio do tipo de imagem típica que se espera adquirir.

Este trabalho apresenta um estudo sobre algumas técnicas modernas de recuperação de imagens e vídeo que sofreram tais degradações, além de propor extensões. Para suprir a falta de informação, são exploradas características de imagens naturais através de modelos (p. ex. deconvolução por *total variation* (TV), interpolação direcional, modelos autorregressivos) e observações adicionais da cena obtidas pelos diversos quadros consecutivos de um vídeo (p. ex. super-resolução e remoção de ruído de padrão fixo).

Os métodos propostos geram resultados encorajadores, indicando que a linha de

pesquisa seguida tem grande potencial. Finalmente, reavaliando-se a literatura mais atual, novas direções e trabalhos futuros são delineados, inspirados em descobertas recentes na área de problemas inversos em processamento de imagens.

1.1 Sobre termos em inglês

Diversos termos em inglês serão usados nesta tese, como *pixel*, *patch*, *aliasing*, *blurring*, *denoising*, etc. Para alguns deles, não há um equivalente em português e/ou não é de costume a tradução. Para outros, considerados palavras chaves, optou-se por não traduzir, pois a vasta maioria da literatura está em inglês e consultas futuras a esses trabalhos poderiam gerar confusão.

1.2 Organização da tese

Ainda sobre questões de língua, boa parte do texto foi escrita em inglês durante doutorado sanduíche na Universidade da Califórnia e também visando a artigos em periódicos e congressos internacionais. Infelizmente, as normas atuais de confecção de trabalhos da COPPE/UFRJ não permitem que o corpo da tese esteja em inglês. Fez-se necessária, então, a escrita do corpo da tese em português. De modo a aproveitar texto já escrito e evitar laboriosas traduções, trechos em inglês se encontram nos anexos, onde são permitidos.

Em termos de conteúdo, a seguinte divisão foi adotada. O corpo da tese em português contém ideias essenciais, discussões e resumos dos anexos. Os anexos em inglês contêm, em geral, detalhes dos desenvolvimentos e as contribuições geradas. Tentou-se escrever os anexos em formato de artigos científicos, ou de maneira que pudessem ser facilmente transformados em artigos.

Embora tal mistura de línguas possa gerar incômodo à leitura, tentou-se minimizar esse efeito conduzindo-se o leitor aos anexos quando necessário e indicando o retorno ao corpo da tese.

1.2.1 Divisão dos capítulos

O Capítulo 2 discute algumas ferramentas clássicas para resolução de problemas inversos e analisa suas aplicação para reconstrução de imagens. São apontadas as limitações dessas abordagens e os *priors* são identificados como as peças que podem ser lapidadas para gerar reconstruções melhores.

No Capítulo 3 (juntamente com Apêndice B, referenciado quando necessário) é apresentado o método de deconvolução de imagens por *total variation*. São apresentadas evidências que suportam seu uso e são discutidos métodos eficientes

Ano	2009	2010		2011		2012	
Atividade\Semestre	2	1	2	1	2	1	2
Capítulo 7 e Apêndice F: Extensão dos algoritmos de remoção de FPN e confecção do artigo [2].	X						
Capítulo 4 e Apêndices C, G, C.4: Estudos iniciais com modelos PAR e defesa do tema de tese		X	X				
Capítulo 3 e Apêndice B: Doutorado Sanduíche - estudo de algoritmos TV e confecção do artigo [3].				X	X		
Capítulo 5 e Apêndices D e E: Estudo e experimentos com *priors*.					X	X	X
Capítulos 1, 2, 6, 8 e Apêndice A: Confecção final da tese.						X	X

Tabela 1.2: Ordem cronológica dos estudos e escrita dos capítulos.

de solução. São propostas extensões que melhoram a qualidade da estimativa final, porém visando à manutenção da eficiência computacional.

No Capítulo 4 (e Apêndice C), modelos autorregressivos por partes (PAR) são apresentados como uma alternativa de regularização localmente adaptativa. A partir de algoritmos de interpolação do estado-da-arte que utilizam modelos PAR, extensões são propostas e começa-se a discutir generalizações dos modelos PAR.

Ainda no Capítulo 4 (e Apêndice C) são efetuados estudos preliminares de aplicações de modelos PAR em super-resolução. A motivação é obter um algoritmo que explore, ao mesmo tempo, regularidades geométricas dentro dos quadros e relações entre quadros que provejam informação adicional sobre a cena.

À luz de novas descoberta na área de problemas inversos, modelos para imagens naturais e *priors* são revisitados no Capítulo 5 e Apêndice E. Por ser uma área de pesquisa muito ativa, são identificadas novas tendências e a linha de pesquisa deste trabalho é realinhada através de propostas de trabalhos futuros no Capítulo 6.

O Capítulo 7 trata de um outro tipo de degradação que tem maiores aplicações em vídeos infravermelhos. Trata-se do ruído de padrão fixo (FPN) que requer métodos distintos de estimação. Por ter sido um trabalho de extensão do mestrado e ser menos relacionado aos outros assuntos desta tese, escolheu-se colocá-lo como último capítulo.

Tentou-se organizar os capítulos numa ordem que privilegiasse a leitura mantivesse o fluxo de ideias mais ou menos constante. Porém, a título de curiosidade, a Tabela 1.2 mostra cronologicamente quando as atividades foram realizadas.

1.3 Contribuições

Destacam-se abaixo as contribuições geradas durante o período de doutorado do candidato.

1. Extensão do trabalho realizado no mestrado sobre correção de ruído de padrão fixo (FPN) [4] com o desenvolvimento do algoritmo *Affine Projection* para estimação e remoção de ruído, detalhado no Apêndice F. Foi submetido e aceito artigo científico a periódico internacional conforme item 1, Seção A.1.

2. Extensão do algoritmo de deconvolução de imagens por *Total Variation* baseado na abordagem *Augmented Lagrangian* e *Alternating Direction Method of Multipliers*. Foi submetido e aceito artigo em congresso internacional conforme item 1, Seção A.2.

3. Extensão do algoritmo de interpolação por modelos autorregressivos e *soft-decision* (SAI: *soft-decision autoregressive interpolation*) para considerar imagem completa e interdependência entre *pixels* distantes, conforme Seção C.2.

4. Extensão do algoritmo descrito no item 3 para realizar super-resolução, ou seja, combinar informações de outros quadros de um vídeo, conforme Seção C.4.

5. Proposta de uso de modelos autorregressivos como regularização localmente adaptativa para problemas de deconvolução de imagem (*deblurring* e interpolação), conforme Seção C.3.

Capítulo 2

Ferramentas clássicas

Este capítulo descreve brevemente o problema de reconstrução de imagens e vídeos, mencionando algumas soluções clássicas e apontando seus problemas. Duas boas referências para o assunto são [5] e [1].

É considerado apenas o caso não-cego, ou seja, o operador direto que gerou a degradação na imagem é, por suposição, conhecido.

2.1 Interpolação bicúbica

Seja uma imagem ou vídeo que teve sua resolução reduzida ou cuja aquisição não tinha a resolução desejada. Representando-se essa operação por

$$\mathbf{g} = \mathbf{R}\mathbf{f} \tag{2.1}$$

não é possível a recuperação da imagem original por

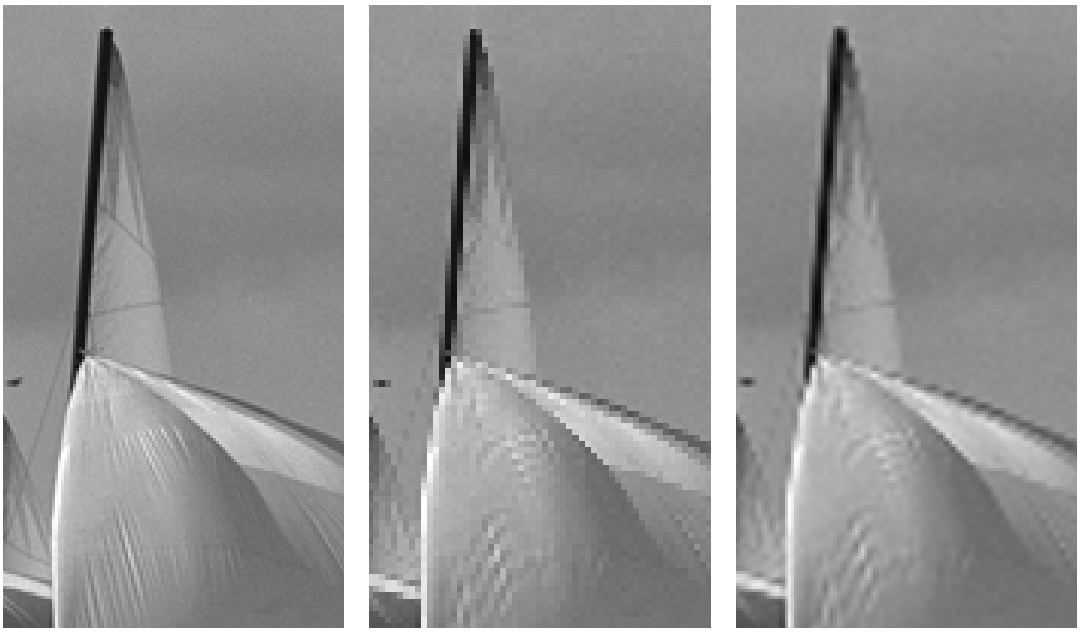
$$\mathbf{f} = \mathbf{R}^{-1}\mathbf{g} \tag{2.2}$$

pois $\mathbf{R}$ não é inversível.

Uma saída é considerar os *pixels* da imagem como amostras de um sinal contínuo e utilizar técnicas de interpolação para aumentar a quantidade de *pixels* na quantidade desejada.

Polinômios são amplamente utilizados para esse fim. A noção mais elementar de interpolação é quando um polinômio é forçado a passar através dos pontos existentes e, após estimar os coeficientes desse polinômio, dados para preencher as lacunas são extraídos da curva contínua. [6].

No entanto, colocar um polinômio de alto grau por todos os pontos não produz bons resultados. Um pequeno movimento de um único ponto pode produzir uma grande alteração no polinômio fazendo-o oscilar violentamente, efeito conhecido como



(a) Imagem original

(b) Imagem subamostrada por 2 nas direções vertical e horizontal

(c) Imagem interpolada por *spline* cúbica

Figura 2.1: Exemplo de *aliasing* causado na subamostragem de uma imagem. Os contornos perdem a definição e a interpolação bicúbica não é capaz de recuperá-los pois assume que a imagem é homogênea, ou seja, não possui estruturas locais.

fenômeno de Runge. Para evitar este problema, pode-se usar uma interpolação por partes, onde diferentes polinômios de baixo grau são usados a cada dois pontos existentes com o cuidado de garantir uma conexão adequada entre eles [7].

Este requisito é atingido pelas *splines*. Os pontos de junção dos polinômios por partes são chamados de *nós*. Para uma *spline* de grau n, cada segmento é um polinômio de grau n, o que sugere que sejam necessários $(n + 1)$ coeficientes para descrever cada sub-polinômio. No entanto, há restrição adicional de suavidade, o que impõe continuidade nos nós até $(n - 1)$-ésima derivada, de maneira que haja apenas um grau de liberdade por segmento [8]. Esta restrição garante que a curva resultante será suave.

A interpolação *bicúbica*, largamente utilizada, emprega superfícies descritas por *splines* de terceiro grau para unir os *pixels* de uma imagem. Desta maneira, é possível se estimar o valor de um *pixel* faltante ou intermediário.

Entretanto, esta forma de interpolação assume que a imagem é homogênea e, portanto, não é capaz de se adaptar a estruturas locais. Com frequência, o resultado final possui contornos mal definidos e borrados, como mostra a Figura 2.1.

Outro problema é que a formulação da *spline* cúbica não leva em conta possível ruído na observação, o que pode afetar negativamente o resultado final da interpolação.

2.2 Filtro inverso e filtro de Wiener

Se não há redução de resolução na observação da imagem, mas há influência da óptica do sistema, borramento, ou outra operação que possa ser modelada por um operador convolucional, i.e.

$$\mathbf{g} = \mathbf{W}\mathbf{f}, \tag{2.3}$$

a recuperação da imagem poderia ser obtida por

$$\mathbf{f} = \mathbf{W}^{-1}\mathbf{g} \tag{2.4}$$

se a matriz $\mathbf{W}$ não for singular. Caso $\mathbf{W}$ seja circulante por blocos, ou seja, represente uma operação de convolução circular de um sistema linear invariante no espaço, a solução por inversa $\mathbf{W}^{-1}$ é equivalente ao filtro inverso.

Neste caso, $\mathbf{W}$ é diagonalizada pela matriz DFT 2-D, ou seja, os autovetores são senóides e seus autovalores representam a resposta em frequência do sistema. A inversa de $\mathbf{W}$ é calculada invertendo-se os autovalores, que equivale à resposta em frequência inversa, ou seja, o filtro inverso.

Caso $\mathbf{W}$ seja singular, pode-se usar a pseudo-inversa, ou inversa generalizada, que é equivalente ou filtro inverso generalizado [5]. Neste caso, os autovalores iguais a zero, ou zeros do filtro, são mantidos, invertendo-se apenas os autovalores diferentes de zero.

O problema desta solução é a instabilidade. Supondo uma observação que contenha ruído, i.e.

$$\mathbf{g} = \mathbf{W}\mathbf{f} + \boldsymbol{\nu} \tag{2.5}$$

e que alguns autovalores de $\mathbf{W}$ sejam muito pequenos, o que é muito comum na prática, a inversa terá ganhos muito elevados e qualquer ruído será demasiadamente amplificado.

Por outro lado, o ruído pode ser combatido por um filtro suavizador que normalmente é um passa-baixas. Porém, como a observação através de $\mathbf{W}$ geralmente atenua altas frequências, não é possível recuperar $\mathbf{f}$ por um filtro passa-baixas.

Essas limitações podem ser solucionadas pelo filtro de Wiener, caso se tenha algum conhecimento sobre o ruído. Basicamente, o filtro de Wiener é um compromisso ótimo entre um filtro suavizador de ruído (passa-baixas) e um filtro inverso (passa-altas).

2.3 Problemas inversos e regularização

Embora o problema direto seja trivial, ou seja, simular uma observação a partir de uma imagem hipotética, o problema inverso normalmente não o é, ver Figura 2.2.

A Figura 2.3 ilustra o problema. Com frequência, o problema inverso de recuperar

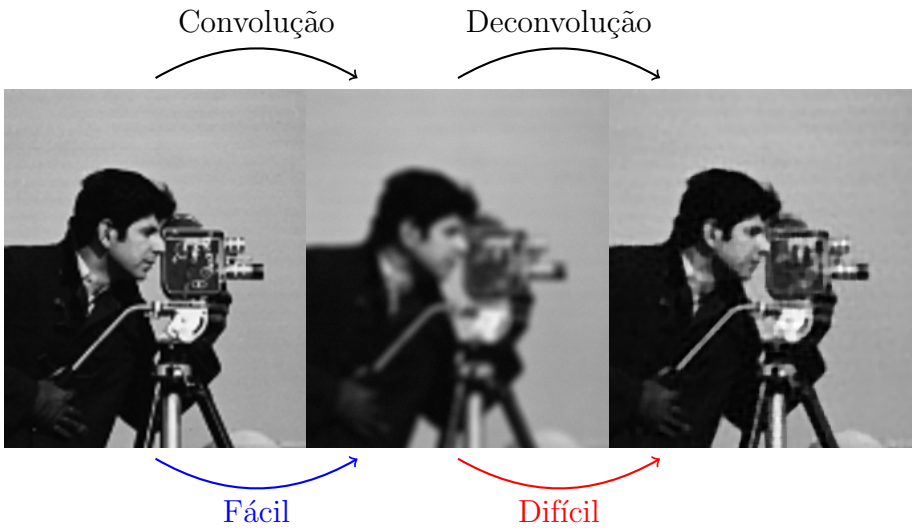


Figura 2.2: O problema direto, que é a simulação de uma observação, é facilmente obtido pela operação de convolução, já que o operador é supostamente conhecido. Neste caso, foi simulada a aquisição de uma imagem fora de foco. Porém, o problema inverso, ou deconvolução, ou recuperação da imagem em foco, é normalmente difícil já que é comum haver perda de informação durante a aquisição do sinal.

a imagem original admite infinitas soluções. Pela interpretação do ponto de vista da álgebra linear, a matriz do operador direto é singular ou mal condicionada. Portanto, é preciso de algum *critério* para se escolher uma solução entre as infinitas possíveis.

2.3.1 Abordagem bayesiana

Um tipo de resposta a esse problema muito usada na literatura é fornecida pela abordagem bayesiana. Suponha-se que a imagem original é uma variável aleatória com função densidade de probabilidade (PDF) $p(\mathbf{f})$. Dada uma observação $\mathbf{g}$ de $\mathbf{f}$, deseja-se a estimativa $\hat{\mathbf{f}}$ que seja a mais provável, ou seja,

$$\hat{\mathbf{f}} = \arg\max_{\mathbf{f}} \; p(\mathbf{f}|\mathbf{g}). \tag{2.6}$$

Usando a regra de Bayes, tem-se

$$p(\mathbf{f}|\mathbf{g}) = \frac{p(\mathbf{g}|\mathbf{f})p(\mathbf{f})}{p(\mathbf{g})}, \tag{2.7}$$

sendo que $p(\mathbf{g})$ pode ser eliminada já que é constante (a variável é $\mathbf{f}$).

Esse problema de maximização pode ser transformado em um problema de minimização

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \; \big\{ -\log p(\mathbf{g}|\mathbf{f}) - \log p(\mathbf{f}) \big\} \tag{2.8}$$

que é conhecido como estimador MAP (*maximum a posteriori*).

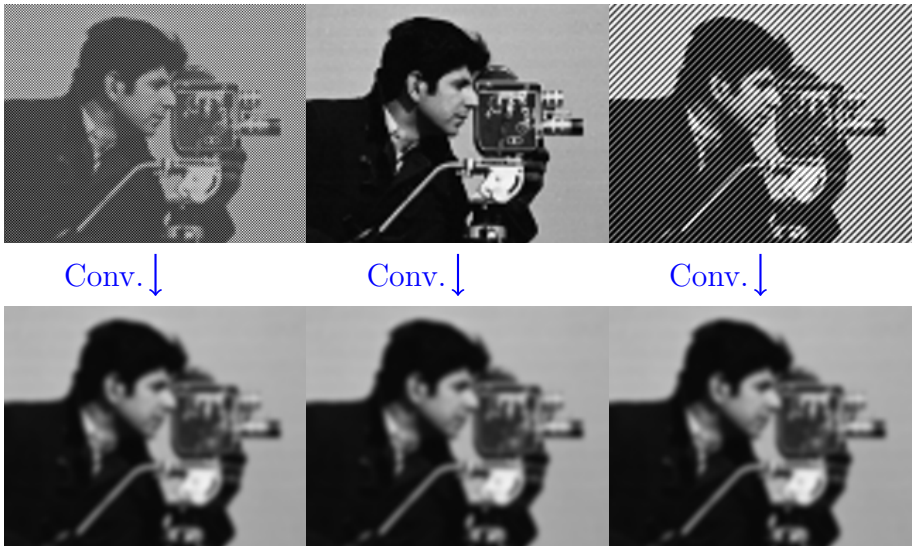


Figura 2.3: Exemplo de três imagens (acima) que quando convoluídas com um operador linear de *blurring* geram a mesma observação (abaixo). Caso só essa observação estivesse disponível, qualquer uma das imagens do topo seria solução para o problema inverso.

O primeiro termo de (2.8), $-\log p(\mathbf{g}|\mathbf{f})$, é facilmente obtido. Como o operador direto é conhecido, $\mathbf{g}$ é facilmente calculado quando $\mathbf{f}$ é dado. Esse termo é chamado de *data fidelity term*, pois mede a probabilidade de uma possível solução $\mathbf{f}$ ter $\mathbf{g}$ como uma observação.

É comum supor que durante a observação a imagem foi contaminada por ruído branco aditivo com distribuição gaussiana (AWGN) com média zero e desvio padrão $\sigma_{\boldsymbol{\nu}}$ e com componentes IID (*independent and identically distributed*). Neste caso, usa-se

$$p(\mathbf{g}|\mathbf{f}) = \frac{1}{\sigma_{\boldsymbol{\nu}}\sqrt{2\pi}} \exp\left\{-\frac{\|\mathbf{W}\mathbf{f}-\mathbf{g}\|^2}{2\sigma_{\boldsymbol{\nu}}^2}\right\} \tag{2.9}$$

quando o modelo de observação é dado por $\mathbf{g} = \mathbf{W}\mathbf{f} + \boldsymbol{\nu}$, pois $\|\mathbf{W}\mathbf{f}-\mathbf{g}\|^2 = \|\boldsymbol{\nu}\|^2$ é apenas ruído. Portanto, é coerente se utilizar uma distribuição normal com o desvio padrão do ruído $\sigma_{\boldsymbol{\nu}}$.

Aplicando-se o logaritmo, o termo que deve ser minimizado é

$$-\log p(\mathbf{g}|\mathbf{f}) = \frac{\|\mathbf{W}\mathbf{f}-\mathbf{g}\|^2}{2\sigma_{\boldsymbol{\nu}}^2} + C, \tag{2.10}$$

onde C é uma constante.

Caso não se tenha nenhuma informação *a priori* sobre a solução, $p(\mathbf{f})$ é constante pois todas as respostas são equiprováveis. Neste caso, a solução obtida por simples mínimos quadrados, conhecida também como solução por ML (*maximum likelihood*)

$$\hat{\mathbf{f}} = (\mathbf{W}^{\mathrm{T}}\mathbf{W})^{-1}\mathbf{W}^{\mathrm{T}}\mathbf{g}. \tag{2.11}$$

O segundo termo, $-\log p(\mathbf{f})$, mede a probabilidade de um dado $\mathbf{f}$ ser solução, independentemente (ou antes) da observação. Neste termo é possível acrescentar algum conhecimento prévio que se tem sobre o problema. Por exemplo, ao se tentar reconstruir imagens naturais[1], deseja-se que $p(\mathbf{f})$ forneça um valor alto se $\mathbf{f}$ for uma imagem natural e baixo caso contrário. Esse termo é comumente chamado de *prior*.

2.3.2 Regularização em problemas inversos

Uma maneira prática de se usar o termo $p(\mathbf{f})$ é obtido pela *regularização* na terminologia de problemas inversos. Caso $\mathbf{W}$ seja mal-condicionada ou singular (como descrito anteriormente), pode-se acrescentar um termo de regularização

$$\varphi(\mathbf{f}), \tag{2.12}$$

que retorna valores baixos caso $\mathbf{f}$ seja uma resposta provável, e altos caso contrário. A formulação do problema é, então, dada por

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \ \frac{\mu}{2} \left\|\mathbf{W}\mathbf{f} - \mathbf{g}\right\|^2 + \varphi(\mathbf{f}), \tag{2.13}$$

onde μ é chamado de *fator de regularização*. Alterando-se μ é possível controlar na resposta final a importância entre fidelidade às observações (*data fidelity*) e concordância com modelos prévios (*prior*).

Outra maneira comum de se escrever 2.13 é

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \ \left\|\mathbf{W}\mathbf{f} - \mathbf{g}\right\|^2 + \lambda\varphi(\mathbf{f}), \tag{2.14}$$

que é equivalente a (2.13) quando $\lambda = {}^{2}/_{\mu}$.

Fazendo $\varphi(\mathbf{f}) = \left\|\mathbf{\Gamma}\mathbf{f}\right\|^2$ obtém-se a regularização Tikhonov, com a solução dada por

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \ \frac{1}{2\sigma_{\boldsymbol{\nu}}^2} \left\|\mathbf{W}\mathbf{f} - \mathbf{g}\right\|^2 + \frac{1}{2} \left\|\mathbf{\Gamma}\mathbf{f}\right\|^2, \tag{2.15}$$

onde $\mathbf{\Gamma}$ é conhecida como matriz de Tikhonov. Embora a definição de $\mathbf{\Gamma}$ seja bem flexível podendo acomodar vários tipos de regularização, para o caso especial $\mathbf{\Gamma} = ({}^{1}/_{\sigma_{\mathbf{f}}})\mathbf{I}$ tem-se

$$\hat{\mathbf{f}} = (\mathbf{W}^{\mathrm{T}}\mathbf{W} + \alpha^2\mathbf{I})^{-1}\mathbf{W}^{\mathrm{T}}\mathbf{g}, \tag{2.16}$$

onde $\alpha = {}^{\sigma_{\boldsymbol{\nu}}}/_{\sigma_{\mathbf{f}}}$.

Do ponto de vista bayesiano, a escolha $\mathbf{\Gamma} = ({}^{1}/_{\sigma_{\mathbf{f}}})\mathbf{I}$ equivale a supor que $\mathbf{f}$ pode

[1] Considera-se imagem natural toda aquela onde os motivos representam alguma faceta da realidade: natureza, pessoas, animais, cidades, etc.

ser descrita por uma distribuição gaussiana com componentes IID tal que

$$p(\mathbf{f}) = \frac{1}{\sigma_{\mathbf{f}}\sqrt{2\pi}} \exp\Big\{-\frac{\mathbf{f}^{\mathrm{T}}\mathbf{f}}{2\sigma_{\mathbf{f}}^2}\Big\}, \tag{2.17}$$

com desvio padrão $\sigma_{\mathbf{f}}$ e média zero. Com isso, quando menor $\|\mathbf{f}\|^2$, mais provável é que $\mathbf{f}$ seja solução. Portanto, (2.16) resulta na solução mínimos quadrados de norma mínima. Comparada a $(\mathbf{W}^{\mathrm{T}}\mathbf{W})^{-1}$ que pode não existir, a inversa $(\mathbf{W}^{\mathrm{T}}\mathbf{W} + \alpha^2\mathbf{I})^{-1}$ é bem definida, já que $\alpha^2\mathbf{I}$ elimina os autovalores iguais a zero.

A solução dada por (2.16) é relacionada ao filtro de Wiener. Supondo que $\mathbf{W}$ tenha a decomposição por valores singulares (SVD)

$$\mathbf{W} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}} \tag{2.18}$$

com valores singulares s_i, (2.16) pode ser re-escrita como

$$\hat{\mathbf{f}} = \mathbf{V}\mathbf{S}'\mathbf{U}^{\mathrm{T}}\mathbf{g}, \tag{2.19}$$

sendo $\mathbf{S}'$ diagonal com elementos

$$[\mathbf{S}']_{ii} = \frac{s_i}{s_i^2 + \alpha^2}. \tag{2.20}$$

Finalmente, os coeficientes do filtro de Wiener são dados por [9]

$$w_i = \frac{s_i^2}{s_i^2 + \alpha^2}. \tag{2.21}$$

2.3.3 Interpolação como problema inverso

O processo de interpolação descrito na Seção 2.1 pode ser interpretado pela abordagem de regularização. Lembremos que a subamostragem pode ser definida por

$$\mathbf{g} = \mathbf{R}\mathbf{f} \tag{2.22}$$

que não tem solução trivial, pois $\mathbf{R}$ não tem inversa. Usando a solução regularizada

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \ \frac{\|\mathbf{R}\mathbf{f} - \mathbf{g}\|^2}{2\sigma_{\boldsymbol{\nu}}^2} + \|\boldsymbol{\Gamma}\mathbf{f}\|^2, \tag{2.23}$$

tem-se

$$\hat{\mathbf{f}} = (\mathbf{R}^{\mathrm{T}}\mathbf{R} + \alpha^2\boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{\Gamma})^{-1}\mathbf{R}^{\mathrm{T}}\mathbf{g}, \tag{2.24}$$

onde $\mathbf{R}^{\mathrm{T}}$ aumenta a dimensão de $\mathbf{g}$ para a dimensão de $\mathbf{f}$ acrescentado zeros (*zero padding*) e $\mathbf{R}^{\mathrm{T}}\mathbf{R}$ atua como uma máscara, zerando os mesmos elementos que são

eliminados por $\mathbf{R}$, porém sem alterar a dimensão do vetor em que atua. Portanto, o termo $\alpha^2\mathbf{\Gamma}^{\mathrm{T}}\mathbf{\Gamma}$ seria responsável por efetivamente preencher os *pixels* faltantes na interpolação.

Usando o conceito de *B-splines* cardinais [8], que são uma generalização da função *sinc* para graus menores, pode-se determinar $\mathbf{\Gamma}$ de tal forma que o termo $(\mathbf{R}^{\mathrm{T}}\mathbf{R}+\alpha^2\mathbf{\Gamma}^{\mathrm{T}}\mathbf{\Gamma})^{-1}$ preencha os *pixels* iguais a zero após $\mathbf{R}^{\mathrm{T}}\mathbf{g}$ respeitando as restrições de continuidade das derivadas exigida pelas *splines*.

Entretanto, essa abordagem seria ineficiente. Pode-se, por exemplo, utilizar uma combinação de filtros recursivos causal e não-causal para o cálculo rápido dos coeficientes das *B-splines* [8].

2.4 Conclusão

Para problemas de reconstrução de imagem, a suposição

$$p(\mathbf{f}) = \frac{1}{\sigma_{\mathbf{f}}\sqrt{2\pi}} \exp\Big\{-\frac{\mathbf{f}^{\mathrm{T}}\mathbf{f}}{2\sigma_{\mathbf{f}}^2}\Big\} \tag{2.25}$$

é muito limitada, pois supõe média zero e matriz de autocovariância de $\mathbf{f}$ igual a $\mathbf{\Sigma_f} = \mathbf{I}$. Ou seja, nenhuma estrutura ou dependência entre os *pixels* de uma imagem é considerada. Por outro lado, é sabido que imagens naturais têm conteúdo em frequência passa-baixas [5, 10–13].

Uma opção seria considerar a matriz de covariância e modelar a imagem como

$$p(\mathbf{f}) = \frac{1}{|\mathbf{\Sigma_f}|^{1/2}\sqrt{2\pi}} \exp\left\{-\frac{1}{2}(\mathbf{f}-\boldsymbol{\mu}_{\mathbf{f}})^{\mathrm{T}}\mathbf{\Sigma}_{\mathbf{f}}^{-1}(\mathbf{f}-\boldsymbol{\mu}_{\mathbf{f}})\right\}, \tag{2.26}$$

que tem solução

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \ \frac{\|\mathbf{W}\mathbf{f}-\mathbf{g}\|^2}{2\sigma_{\boldsymbol{\nu}}^2} + \frac{\left\|\mathbf{\Sigma}_{\mathbf{f}}^{-1/2}(\mathbf{f}-\boldsymbol{\mu}_{\mathbf{f}})\right\|^2}{2}, \tag{2.27}$$

podendo ser obtida por

$$\hat{\mathbf{f}} = \left(\mathbf{W}^{\mathrm{T}}\mathbf{W} + \sigma_{\boldsymbol{\nu}}^2\mathbf{\Sigma}_{\mathbf{f}}^{-1}\right)^{-1} \cdot \left(\mathbf{W}^{\mathrm{T}}\mathbf{g} + \mathbf{\Sigma}_{\mathbf{f}}^{-1}\boldsymbol{\mu}_{\mathbf{f}}\right). \tag{2.28}$$

Mas como será visto, imagens têm comportamento não-gaussiano e tal solução tende a gerar resultados pobres.

Em suma, abordagens clássicas não são boas alternativas para reconstrução de imagens, seja no caso de interpolação, *deblurring* ou *denoising*, pois não modelam $p(\mathbf{f})$ adequadamente. Estruturas típicas de imagens naturais, como contornos, texturas,

etc., são ignoradas, já que se supõe que as imagens são homogêneas.

A pesquisa de problemas inversos em processamento de imagens tem o foco numa melhor definição de $p(\mathbf{f})$ aliada a métodos eficientes de otimização. Um bom modelo para imagens pode implicar soluções impraticáveis em termos de otimização; soluções eficientes podem gerar resultados pobres. Logo, o compromisso entre ambos os aspectos é a resposta.

Nos próximos capítulos serão abordadas alternativas de *priors*, ou seja, modelos mais adequados para imagens naturais.

Capítulo 3

Deconvolução por Total Variation

Um dos *priors* de maior sucesso para solução de problemas inversos em processamento de imagens é através do uso de *Total Variation* (TV) como regularização. O sucesso de abordagem por TV é atribuído a um bom compromisso entre habilidade de descrever imagens suaves por partes (sem penalizar descontinuidades) e complexidade dos algoritmos [12, 14, 15].

Este capítulo descreve sua formulação, aponta métodos eficientes de cálculo e propõe alterações que melhoram a qualidade da imagem final sem comprometer a rapidez do algoritmo.

Introdução

Como já mencionado, é possível melhorar a solução de problemas inversos com uma melhor descrição ou modelo do sinal em questão. Tomando-se como exemplo um problema de *deblurring* e lembrando que imagens naturais têm conteúdo passa-baixas, pode-se querer penalizar altas frequências durante o processo de reconstrução através de filtros passa-altas tipo derivadas, ou seja,

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \ \frac{\mu}{2} \left\| \mathbf{H}\mathbf{f} - \mathbf{g} \right\|^2 + \beta_x \left\| \mathbf{D}_x\mathbf{f} \right\|^2 + \beta_y \left\| \mathbf{D}_y\mathbf{f} \right\|^2 , \tag{3.1}$$

onde as matrizes $\mathbf{D}_x$ e $\mathbf{D}_y$ são construídas para realizar derivadas horizontais e verticais e as constantes β_x e β_y controlam o teor de regularização na horizontal e vertical respectivamente.

Através de (3.1), imagens com grande variação entre os *pixels* e altas frequências serão preteridas. Embora conceitualmente interessante, tal abordagem gera, infelizmente, resultados pobres com imagens demasiadamente suaves (*oversmoothed*) ou com comportamente oscilatório. O problema, como será visto, está na estatística do termo de regularização.

Diversas imagens foram consideradas em um estudo estatístico em [16, 17]. O

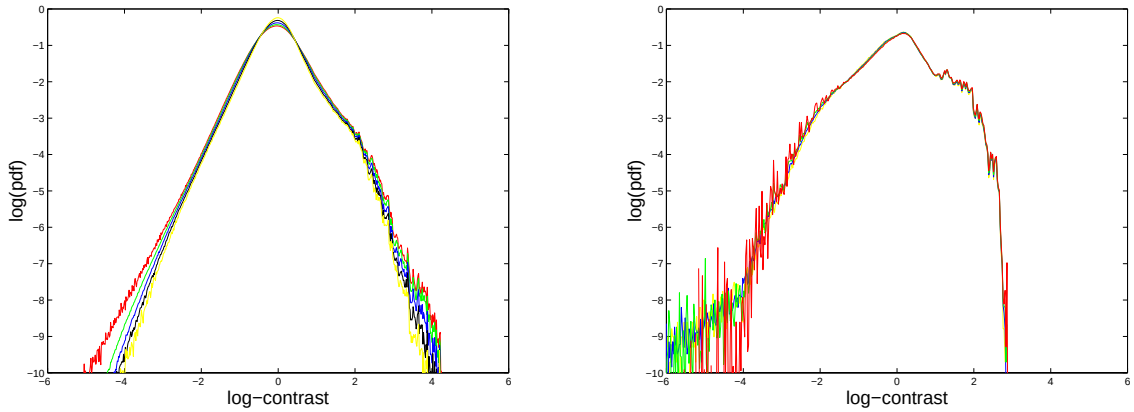


Figura 3.1: Logaritmo da PDF dos *pixels* considerados individualmente, de diferentes banco de dados (diferentes figuras) em diferentes escalas (cores) [16]

objetivo foi examinar como os parâmetros estatísticos variavam entre diferentes bancos de dados de imagens e em diferentes escalas[1].

A primeira conclusão foi que, quando considerados individualmente, os *pixels* de imagens naturais têm uma distribuição não-gaussiana. Os parâmetros avaliados (média, desvio padrão, curtose, etc.) permanecem relativamente invariantes para diferentes escalas da mesma imagem. Porém, comparando resultados de diferentes bancos de dados de imagens, a conclusão foi que os parâmetros são muito instáveis e podem ser consideravelmente diferentes entre diferentes banco de dados, ver Figura 3.1.

Por outro lado, quando os mesmos parâmetros foram avaliados nas derivadas horizontais e verticais das imagens[2], os parâmetros estatísticos permaneceram substencialmente mais consistentes entre diferentes bancos de dados e entre diferentes escalas, ver Figura 3.2.

É de suma importância que parâmetros estatísticos variem pouco entre diferentes grupos de imagens, pois deseja-se ultimamente um *prior* que seja o mais genérico possível (em termos de imagens naturais) ao mesmo tempo que forneça uma boa descrição do sinal.

Reanalisando, agora, a solução proposta na equação (3.1), percebe-se que a limitação está na suposição de que as derivadas da imagem têm distribuição gaussiana, enquanto que na realidade possuem distribuição não-gaussiana.

Outra interpretação é que o *prior* adotado penaliza demasiadamente os contornos (*edges*), uma característica fundamental de imagens naturais que nos possibilita distinguir diferentes objetos. Isso ocorre pois a PDF gaussiana não possui caudas suficientemente longas, o que faz que as grandes variações de intensidade presentes

[1]Duas maneiras de se obter outra escala foram consideradas no estudo: (1) extraindo-se a parte central das imagens e (2) subamostrando as imagens após filtros *anti-aliasing*.

[2]No caso de imagens digitais, as derivadas são aproximadas pelas diferenças entre *pixels* adjacentes.

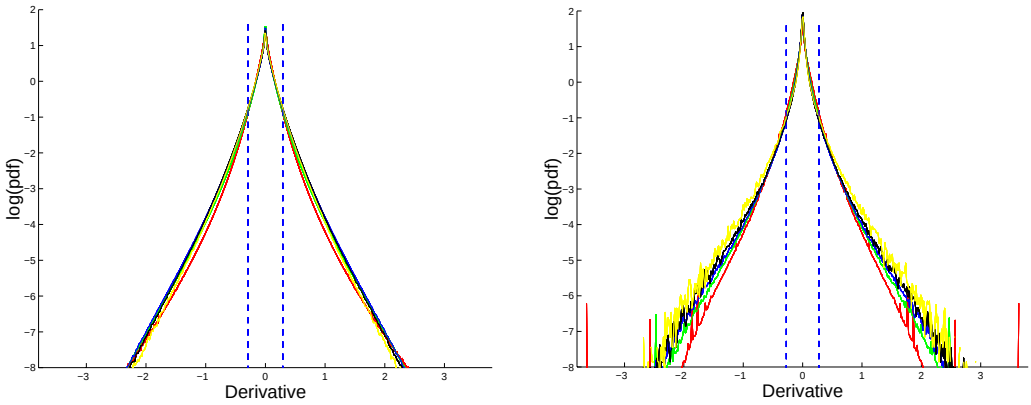


Figura 3.2: Logaritmo da PDF das derivadas das imagens de diferentes banco de dados (diferentes figuras) em diferentes escalas (cores) [16]

nos contornos sejam consideradas muito improváveis, quando não verdade não são.

Para se estudar distribuições ditas não-gaussianas, pode-se utilizar a distribuição gaussiana generalizada (GG) dada por

$$p(x;\mu,\sigma,\phi) = \frac{1}{Z}\exp\left\{-\left|\frac{x-\mu}{A(\phi,\sigma)}\right|^{\phi}\right\}, \tag{3.2}$$

com

$$A(\phi,\sigma) = \left[\frac{\sigma^2\Gamma(1/\phi)}{\Gamma(3/\phi)}\right]^{1/2} \tag{3.3}$$

e

$$Z = 2\Gamma\left(1+1/\phi\right)A(\phi,\sigma), \tag{3.4}$$

onde Z é chamada de *partition function* e tem a função de normalização tal que $\int p\,\mathrm{d}x = 1$, x é a variável aleatória e ϕ é o parâmetro que controla o formato da distribuição de maneira independente da média e da variância. Quando $\phi = 2$, por exemplo, tem-se a distribuição gaussiana clássica. Para $\phi = 1$ a distribuição se torna laplaciana e quando $\phi \to \infty$ obtém-se a distribuição uniforme. A Figura 3.3 mostra alguns exemplos da distribuição gaussiana generalizada para alguns valores de ϕ. Um método prático para se estimar o valor de ϕ a partir de uma amostra é descrito no Apêndice H.

Em [16], os parâmetros de formato (*shape parameter*) ϕ estimados para os diversos banco de dados e diversas escalas variam entre 0,58 e 0,73, ou seja, a distribuição das derivadas está mais próxima da laplaciana. Outros estudos também indicam valores de ϕ entre 0,5 e 0,8 [18, 19].

Duas características da distribuição laplaciana se adequam ao resultado obtido: (1) pico em torno do zero, que está de acordo com o fato de imagens serem suaves, logo derivadas com valor baixo e (2) cauda longa (mais longa que a gaussiana), provenientes dos contornos onde há descontinuidades, ou seja, derivadas com valores

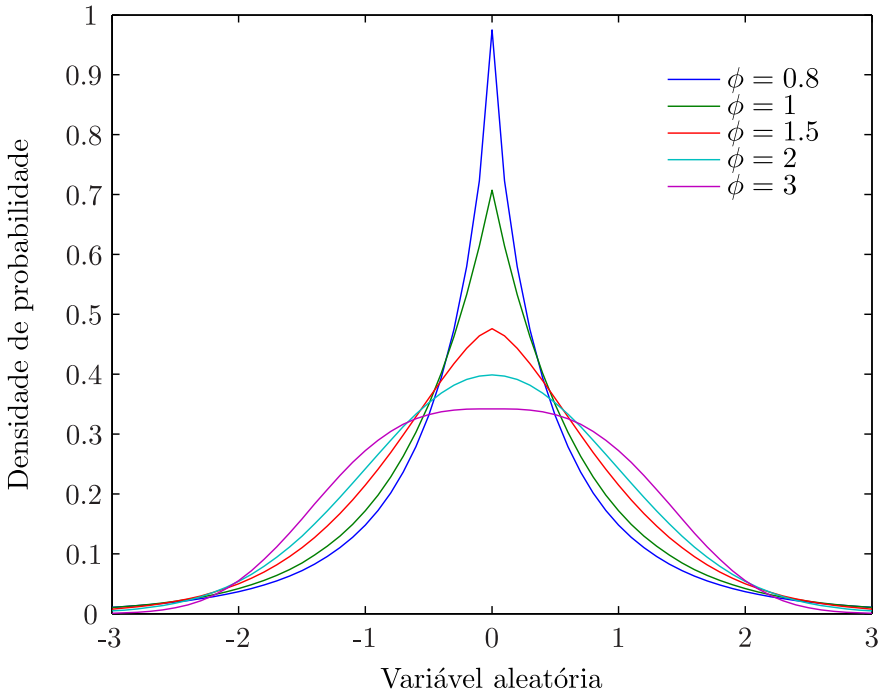


Figura 3.3: Exemplos de PDF da distribuição gaussiana generalizada para diferentes valores do parâmetro de formato ϕ.

altos. Em suma, a distribuição laplaciana caracteriza bem funções suaves por partes com possíveis e permitidas descontinuidades.

Seguindo essa linha de raciocínio, Rudin *et al.* [20] propuseram o uso de *Total Variation* como *prior* em problemas inversos envolvendo imagens. Em sua forma contínua multivariável é definido como

$$\mathrm{TV}(f) = \int_{\Omega} |\nabla f(\mathbf{x})| \, \mathrm{d}\mathbf{x}, \tag{3.5}$$

onde $\mathbf{x} \in \mathbf{R}^n$ e $\Omega \subseteq \mathbf{R}^n$ é o domínio de integração. Intuitivamente, quanto mais oscilante for a função, maior será sua "variação total". Para imagens naturais que têm conteúdo em frequência passa-baixas, é esperado que essa medida seja baixa.

No caso de imagens digitais, é adotada uma versão discreta, chamada de TV isotrópico, e o termo de regularização é dado por

$$\varphi(\mathbf{f}) \rightarrow \|\mathbf{f}\|_{\mathrm{TV2}} = \sum_i \sqrt{\beta_x^2 \left[\mathbf{D}_x \mathbf{f}\right]_i^2 + \beta_y^2 \left[\mathbf{D}_y \mathbf{f}\right]_i^2}. \tag{3.6}$$

Note-se que $\|\mathbf{f}\|_{\mathrm{TV2}} \neq \|\mathbf{D}\mathbf{f}\|_2$, com $\mathbf{D} = [\beta_x \mathbf{D}_x^{\mathrm{T}}, \ \beta_y \mathbf{D}_y^{\mathrm{T}}]^{\mathrm{T}}$ um operador composto de dois suboperadores. Será usada a definição $\|\mathbf{f}\|_{\mathrm{TV2}} \stackrel{\mathrm{def}}{=} \|\mathbf{D}\mathbf{f}\|_{\mathrm{ISO}}$ por ser uma notação mais compacta.

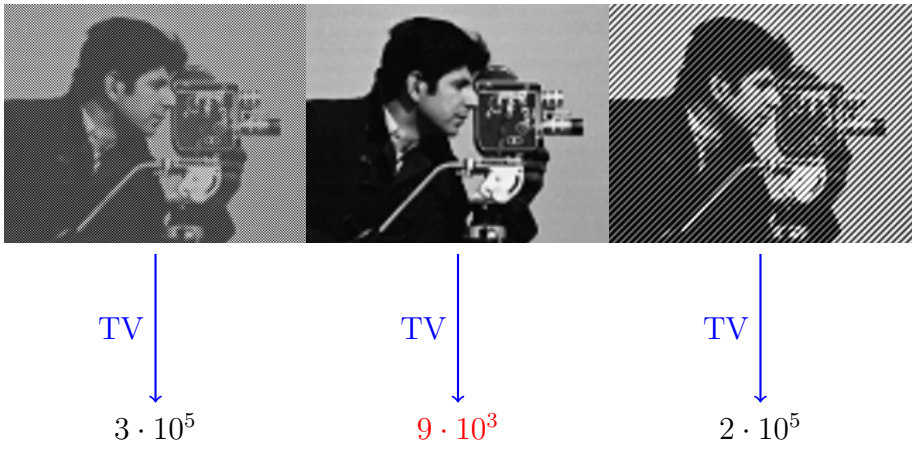


Figura 3.4: Exemplo do uso de TV como medida de seleção da reposta mais apropriada para o problema da Figura 2.3. Sem a regularização, as três imagens são solução para o problema. Porém, a imagem central possui o menor valor de "variação total".

Usando-o como *prior*, a solução do problema é dada por

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \frac{\mu}{2} \|\mathbf{H}\mathbf{f} - \mathbf{g}\|^2 + \|\mathbf{D}\mathbf{f}\|_{\text{ISO}} \,. \tag{3.7}$$

Assim, a solução final será um balanço entre o mínimo de TV e o mínimo de desacordo às observações. A Figura 3.4 ilustra o funcionamento do uso do TV para o exemplo de *deblurring* apresentado na Figura 2.3.

Um das limitações do TV isotrópico, inicialmente proposto em [20], é o arredondamento de cantos e quinas (*corners*) quando usado como *prior*. Uma alternativa é usar no seu lugar o TV anisotrópico, definido como

$$\|\mathbf{f}\|_{\text{TV1}} = \sum_i \beta_x \left|[\mathbf{D}_x\mathbf{f}]_i\right| + \beta_y \left|[\mathbf{D}_y\mathbf{f}]_i\right| = \|\mathbf{D}\mathbf{f}\|_1 \,, \tag{3.8}$$

que neste caso tende a manter cantos horizontais e verticais. Em [21], por exemplo, este tipo de *prior* foi usado para *deblurring* e *denoising* de imagens de códigos de barras 2-D. Outros usos e extensões de TV anisotrópico podem ser encontradas em [22–24].

As Figuras 3.5, 3.6 e 3.7 ilustram as diferenças entre a norma ℓ_2, TV anisotrópico e TV isotrópico. Pode-se perceber que no TV isotrópico todas as direções são igualmente penalizadas e, por esse motivo, tende a fornecer melhores resultados para imagens genéricas. Neste trabalho será adotado TV isotrópico.

Do ponto de vista bayesiano, TV isotróprico supõe

$$p(\mathbf{f}) = \frac{1}{Z} \exp\left\{ -\sum_i \sqrt{\beta_x^2 \left[\mathbf{D}_x\mathbf{f}\right]_i^2 + \beta_y^2 \left[\mathbf{D}_y\mathbf{f}\right]_i^2} \right\}. \tag{3.9}$$

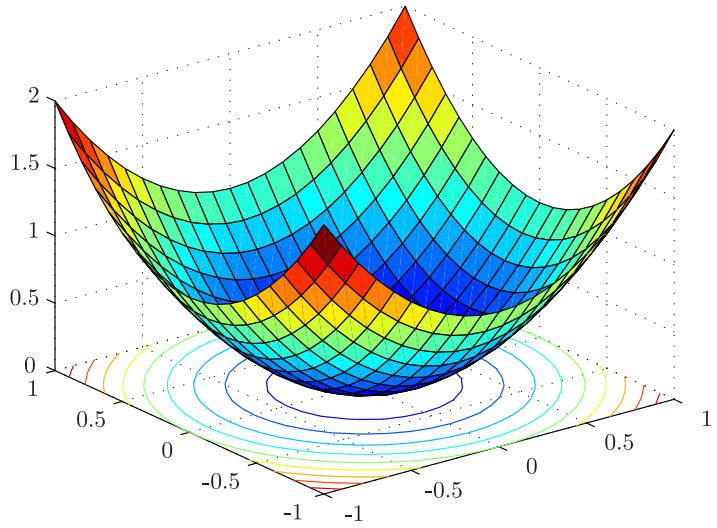


Figura 3.5: Gráfico da norma ℓ_2: $\|\cdot\|^2$

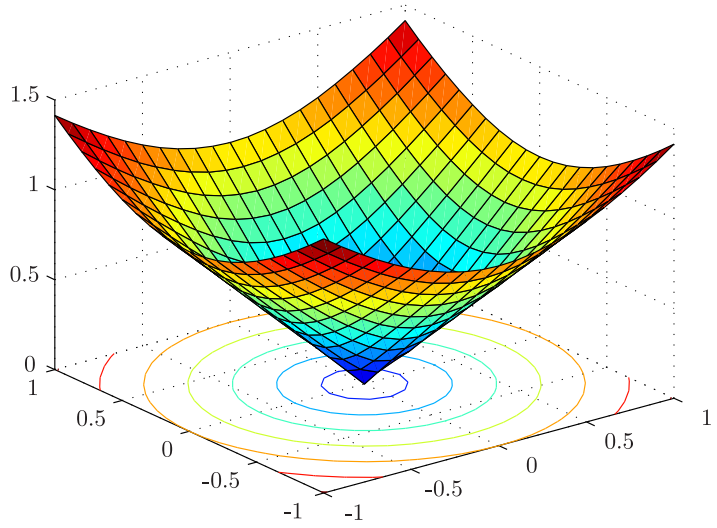


Figura 3.6: Gráfico do TV isotrópico: $\|\cdot\|_{\mathrm{TV2}}$

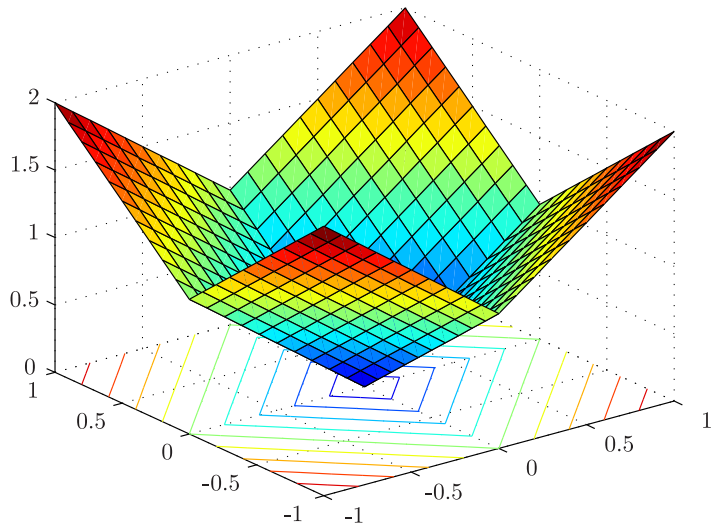


Figura 3.7: Gráfico do TV anisotrópico: $\|\cdot\|_{\mathrm{TV1}}$

3.1 Solução por Augmented Lagrangian

A suposição que as variáveis podem ser descritas por distribuições gaussianas implica que a solução é dada por um estimador linear com soluções rápidas e bem conhecidas [25].

Infelizmente, o regularizador TV supõe distribuição laplaciana que, por envolver norma ℓ_1 e não ser diferenciável, dificulta a solução do problema por métodos convencionais, p. ex. gradiente. É preciso, então, recorrer a alternativas. Diversas estratégias já foram propostas para superar esse fato [14, 20, 26–32].

Recentemente, diversos algoritmos rápidos que compartilham a mesma idéia foram propostos com nomes distintos [33, 34]: *split Bregman iterations* [35], *iterative shrinkage algorithm* [36, 37], *alternating direction method of multipliers* [31] e *majorization–minimization algorithm* [14, 38]. Neste trababalho será adotada a abordagem *Augmented Lagrangian*, também conhecida como *Method of Multipliers* [31, 39–42].

3.1.1 Descrição do método

Basicamente, a rapidez na solução é atingida dividindo-se o problema da equação (3.7) em uma série de problemas mais simples e que possuem solução rápida.

Primeiramente, introduz-se a variável intermediária $\mathbf{u}$ e transforma-se o problema de otimização sem restrições em um problema equivalente com restrições dado por

$$\begin{aligned} &\underset{\mathbf{f},\mathbf{u}}{\text{minimizar}} \ \frac{\mu}{2}\left\|\mathbf{Hf}-\mathbf{g}\right\|^2 + \left\|\mathbf{u}\right\|_{\text{ISO}} \\ &\text{sujeito a } \ \mathbf{u}=\mathbf{Df}. \end{aligned} \tag{3.10}$$

Em seguida, define-se o *augmented Lagrangian* (AL) [31, 42, 43]

$$L(\mathbf{f},\mathbf{u},\mathbf{y}) = \frac{\mu}{2}\left\|\mathbf{Hf}-\mathbf{g}\right\|^2 + \left\|\mathbf{u}\right\|_{\text{ISO}} - \mathbf{y}^{\text{T}}(\mathbf{u}-\mathbf{Df}) + \frac{\rho}{2}\left\|\mathbf{u}-\mathbf{Df}\right\|^2, \tag{3.11}$$

onde ρ é o fator de regularização associado à penalização quadrática $\left\|\mathbf{u}-\mathbf{Df}\right\|^2$, e $\mathbf{y}$ é o multiplicador de Lagrange associado à restrição $\mathbf{u}=\mathbf{Df}$.

A ideia do método é achar um ponto estacionário em $L(\mathbf{f},\mathbf{u},\mathbf{y})$ que também seja solução do problema original (3.7). Para tanto, o *alternating direction method of multipliers* (ADMM) pode ser usado para resolver iterativamente os seguintes sub-problemas [31, 42, 44]:

$$\hat{\mathbf{f}}_{q+1} = \arg\min_{\mathbf{f}} \ \frac{\mu}{2}\left\|\mathbf{H}\mathbf{f}-\mathbf{g}\right\|^2 - \hat{\mathbf{y}}_q^{\mathrm{T}}(\hat{\mathbf{u}}_q - \mathbf{D}\mathbf{f}) + \frac{\rho}{2}\left\|\hat{\mathbf{u}}_q - \mathbf{D}\mathbf{f}\right\|^2 \tag{3.12}$$

$$\hat{\mathbf{u}}_{q+1} = \arg\min_{\mathbf{u}} \ \left\|\mathbf{u}\right\|_{\mathrm{ISO}} - \hat{\mathbf{y}}_q^{\mathrm{T}}(\mathbf{u} - \mathbf{D}\hat{\mathbf{f}}_{q+1}) + \frac{\rho}{2}\left\|\mathbf{u} - \mathbf{D}\hat{\mathbf{f}}_{q+1}\right\|^2 \tag{3.13}$$

$$\hat{\mathbf{y}}_{q+1} = \hat{\mathbf{y}}_q - \rho(\hat{\mathbf{u}}_{q+1} - \mathbf{D}\hat{\mathbf{f}}_{q+1}). \tag{3.14}$$

Finalmente, o subproblema $\mathbf{f}$ em (3.12) tem solução fechada e pode ser calculado eficientemente por FFT [42], pois só envolve norma ℓ_2 e matrizes circulantes por blocos.

O subproblema $\mathbf{u}$ em (3.13) pode ser resolvido usando *shrinkage formula* [45–47] com baixo consumo computacional. Finalmente, o subproblema $\mathbf{y}$ consiste de uma mera atualização. Detalhes e provas do algoritmo serão apresentadas na Seção B.7 do Apêndice B.

3.1.2 Entendendo a abordagem por Augmented Lagrangian

Em geral, problemas de otimização sem restrições são mais fáceis de resolver que problemas com restrições. Logo, reescrever (3.7) como (3.10) não parece ser uma boa alternativa.

Considere-se, a princípio, a abordagem por penalização quadrática sem multiplicadores de Lagrange

$$L(\mathbf{f},\mathbf{u}) = \frac{\mu}{2}\left\|\mathbf{H}\mathbf{f}-\mathbf{g}\right\|^2 + \left\|\mathbf{u}\right\|_{\mathrm{ISO}} + \frac{\rho}{2}\left\|\mathbf{u}-\mathbf{D}\mathbf{f}\right\|^2. \tag{3.15}$$

A vantagem dessa abordagem comparada à otimização sem restrições definida inicialmente em (3.7) é que o termo que envolve norma ℓ_1 não diferenciável (*prior*) foi isolado do termo de *data fidelity* [43]. Dessa maneira, de posse de uma primeira estimativa para $\mathbf{u}$, é possível estimar $\mathbf{f}$ por algoritmos de otimização convencionais.

O passo seguinte seria estimar $\mathbf{u}$, o que envolveria um problema de mínimos quadrados com restrição na norma ℓ_1 dos coeficientes, que poderia ser resolvido pelo algoritmo LASSO [48], e repetir iterativamente o esquema até a convergência. A essa técnica, que foi usada recentemente em vários trabalhos envolvendo processamento de imagens [35, 49–53], dá-se o nome de *variable splitting*.

O problema dessa abordagem é que o coeficiente de penalização ρ teria de ser incrementado sucessivamente de modo que o peso fosse suficiente para garantir que a restrição de igualdade $\mathbf{u} = \mathbf{D}\mathbf{f}$ fosse satisfeita, podendo ter que atingir valores muito elevados. A escolha da sequência de valores para ρ não é uma tarefa fácil, pois pode levar à instabilidade e não-convergência do problema [43].

Considerando agora a abordagem por multiplicadores de Lagrange sem penalização

quadrática, tem-se o Lagrangiano de (3.10) como

$$L(\mathbf{f}, \mathbf{u}, \mathbf{y}) = \frac{\mu}{2} \|\mathbf{Hf} - \mathbf{g}\|^2 + \|\mathbf{u}\|_{\text{ISO}} - \mathbf{y}^{\text{T}}(\mathbf{u} - \mathbf{Df}). \tag{3.16}$$

O problema dessa abordagem é que uma solução para (3.16) é um ponto estacionário de (3.7) e não necessariamente um mínimo. Para ser um mínimo, condições envolvendo derivadas segundas teriam que ser levadas em conta.

A abordagem por *augmented Lagrangian* (repetida por conveniência)

$$L(\mathbf{f}, \mathbf{u}, \mathbf{y}) = \frac{\mu}{2} \|\mathbf{Hf} - \mathbf{g}\|^2 + \|\mathbf{u}\|_{\text{ISO}} - \mathbf{y}^{\text{T}}(\mathbf{u} - \mathbf{Df}) + \frac{\rho}{2} \|\mathbf{u} - \mathbf{Df}\|^2, \tag{3.17}$$

por outro lado, combina as abordagens por penalização quadrática e por multiplicadores de Lagrange de modo a superar suas limitações individuais, fornecendo duas maneiras de melhorar a acurácia da resposta. Como consequência, é possível obter boas aproximações para resposta do problema sem ter de aumentar indefinidamente o termo de penalização quadrática ρ.

O uso conjugado das duas técnicas permite, a cada iteração, estimar explicitamente os multiplicadores de Lagrange $\mathbf{y}$ que, de fato, indicam o peso correto para balancear os gradientes da função objetivo e da restrição de igualdade $\mathbf{u} = \mathbf{Df}$. Também é possível, e recomendado, atualizar o valor de ρ para acelerar ainda mais a convergência [40–43].

Em relação ao método tradicional de multiplicadores de Lagrange, teoremas garantem que uma solução para o *augmented Lagrangian* é, de fato, um mínimo de (3.7) e não apenas um ponto estacionário. Provas e justificativas mais detalhadas do uso dessa abordagem podem ser encontradas em [43] e [54]. A convergência do esquema ADMM é estudada em [35, 55].

3.2 Método proposto: deconvolução por Total Variation através de decomposições direcionais

Uma característica importante não é levada em conta pelo TV convencional: o fato de imagens naturais não serem estacionárias. Basicamente, isso indica que seria mais apropriado aplicar diferentes *priors* em diferentes regiões das imagens.

Uma opção seria dividir a imagens em blocos e resolver o problema por partes, já que pequenos blocos de imagens naturais têm conteúdo aproximadamente estacionário. Quando isso é feito, entretanto, há problemas nas fronteiras dos blocos, podendo o resultado final ficar com o aspecto quadriculado.

Embora existam alternativas para evitar esse fenômeno, p. ex. utilizar blocos maiores com sobreposição (*overlap*), propõe-se aqui considerar a imagem como um

todo, porém tentando adaptar o tipo de *prior* a diferentes regiões da imagem.

Conduz-se o leitor, neste momento, ao Apêndice B, página 49, onde o método proposto é apresentado. A motivação é atenuar os problemas do TV tradicional sem comprometer demasiadamente o desempenho do algoritmo. O apêndice é uma versão estendida do artigo [3].

Capítulo 4

Regularização por modelos autorregressivos

Neste capítulo será investigado o uso de modelos autorregressivos por partes (PAR - *piece-wise autoregressive*) como regularização em problemas inversos. A ideia é ter uma descrição localmente adaptativa de modo a melhorar o *prior* para problemas de reconstrução de imagens.

4.1 Introdução

Como mencionado anteriormente, imagens possuem conteúdo rico e altamente não-estacionário. Porém, é comum aproximá-las para um processo estacionário em sentido amplo (WSS - *wide-sense stationary*) localmente ou quando dividida em pequenos blocos.

Umas das vantagens dessa aproximação é poder descrever imagens por modelos autorregressivos (AR), que por serem locais e variarem espacialmente ao longo da imagem, são denominados "por partes", ou *piece-wise autoregressive* (PAR).

Basicamente, assume-se que é possível estimar o valor de um determinado *pixel* através de uma combinação linear de seus vizinhos[1], ou seja,

$$f(\mathbf{n}) = \sum_{\mathbf{m}\in\mathcal{T}} \alpha_m(\mathbf{n}) f(\mathbf{n}+\mathbf{m}) + \nu(\mathbf{n}), \tag{4.1}$$

onde $\mathbf{n}$ é a coordenada de um determinado *pixel*, $\mathcal{T}$ define sua vizinhança, $\alpha_m(\mathbf{n})$ são os coeficientes do modelo, m é o índice de $\mathbf{m} \in \mathcal{T}$ e $\nu(\mathbf{n})$ é ruído aditivo.

[1]Considera-se que, para imagens, as denominações "causal" e "não-causal" são irrelevantes, já que todos os *pixels* estão disponíveis.

A vizinhança pode ser definida, por exemplo, como

$$\begin{aligned}\mathcal{T} = \{ \quad & [-1,-1]^{\mathrm{T}}, \quad [-1,0]^{\mathrm{T}}, \quad [-1,1]^{\mathrm{T}}, \\ & [0,-1]^{\mathrm{T}}, \qquad\qquad [0,1]^{\mathrm{T}}, \\ & [1,-1]^{\mathrm{T}}, \qquad [1,0]^{\mathrm{T}} \qquad [1,1]^{\mathrm{T}} \quad \}.\end{aligned} \tag{4.2}$$

Supondo que se tem acesso à imagem original, pode-se estimar os parâmetros do modelo através de

$$\hat{\boldsymbol{\alpha}}(\mathbf{n}) = \arg\min_{\boldsymbol{\alpha}} \sum_{\mathbf{w}\in W} \left[f(\mathbf{n}+\mathbf{w}) - \sum_{\mathbf{m}\in\mathcal{T}} \alpha_m f(\mathbf{n}+\mathbf{w}+\mathbf{m}) \right]^2 \tag{4.3}$$

com, por exemplo,

$$\begin{aligned}W = \{ \quad & [-1,-1]^{\mathrm{T}}, \quad [-1,0]^{\mathrm{T}}, \quad [-1,1]^{\mathrm{T}}, \\ & [0,-1]^{\mathrm{T}}, \qquad [0,0]^{\mathrm{T}} \qquad [0,1]^{\mathrm{T}}, \qquad , \\ & [1,-1]^{\mathrm{T}}, \qquad [1,0]^{\mathrm{T}} \qquad [1,1]^{\mathrm{T}} \quad \}\end{aligned} \tag{4.4}$$

chamada de vizinhança, ou janela de treinamento, onde os coeficientes do modelos são treinados.

Uma maneira compacta e conveniente de escrever (4.1), que será adotada quando possível, é

$$\mathbf{f} = \mathbf{A}\mathbf{f} + \boldsymbol{\nu}, \tag{4.5}$$

onde $\mathbf{A}$ é uma matriz com diagonal igual a zero que engloba todos os coeficientes dos modelos autorregressivos, potencialmente diferentes para cada *pixel* da imagem, e $\boldsymbol{\nu}$ é um vetor aleatório chamado, na nomenclatura de modelos AR, de excitação do processo. Um modelo semelhante foi usado recentemente em [56] e [57].

Supondo que os coeficientes do modelo são conhecidos, pode-se definir um *prior* como

$$p(\mathbf{f}) = \frac{1}{Z} \exp\left\{ -\frac{\|\mathbf{f} - \mathbf{A}\mathbf{f}\|_\phi^\phi}{S} \right\} \tag{4.6}$$

$$= \frac{1}{Z} \exp\left\{ -\frac{\|(\mathbf{I} - \mathbf{A})\,\mathbf{f}\|_\phi^\phi}{S} \right\}, \tag{4.7}$$

onde ϕ é o tipo da norma usada e S é um coeficiente de ajuste. Esse *prior* mede, portanto, a probabilidade de uma dada imagem $\mathbf{f}$ ser bem modelada por $\mathbf{A}$.

4.2 Estudos e desenvolvimentos preliminares

Modelos PAR foram empregados com sucesso em interpolação de imagens em [58] onde o objetivo foi privilegiar contornos. Esse trabalho foi, de certa forma, estendido em [59], onde um esquema mais robusto foi adotado. Esse último, por sua vez, foi estendido em [60] que considera o uso de *weighted least squares* para estimar os parâmetros do modelo e melhorar os resultados ainda mais. Outros algoritmos inspirados em [59] podem ser encontrados em [61–65].

De uma maneira genérica, a solução de um problema inverso utilizando-se modelos PAR como *prior* é simples e pode ser dada por

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \ \|\mathbf{W}\mathbf{f} - \mathbf{g}\|^2 + \lambda \|(\mathbf{I} - \mathbf{A})\,\mathbf{f}\|_{\phi}^{\phi}. \tag{4.8}$$

Contudo, o desafio maior está em como a matriz $\mathbf{A}$ é obtida, e o sucesso ou fracasso de um método está na qualidade da estimativa do modelo.

Os estudos iniciais utilizando modelos PAR para interpolação de imagens foram realizados durante o exame de qualificação do candidato e basearam-se no algoritmo descrito em [59]. Embora proposto em 2008, esse método, denominado SAI (*Soft-decision Autoregressive Interpolation*) ainda produz resultados considerados estado-da-arte em termos de interpolação de imagens [66].

O método SAI foi estudado e extensões foram propostas de modo a melhorar os resultados do algoritmo original e aplicá-lo a outros tipos de problemas.

Super-resolução

Também foi estudado o uso de modelos PAR para realizar super-resolução. Neste trabalho, será usado o termo "super-resolução" para a técnica de reconstrução de quadros de um vídeo supondo que os quadros são diferentes observações da mesma cena [67]. O termo *multi-frame super-resolution* também será usado para enfatizar o uso de vários quadros.

Basicamente, tem-se

$$\mathbf{g}_k = \mathbf{R}\mathbf{H}\mathbf{M}_k\mathbf{f} + \boldsymbol{\nu}, \tag{4.9}$$

onde $\mathbf{g}_k$ são várias observações de $\mathbf{f}$ e $\mathbf{M}_k$ são matrizes que modelam tanto deslocamentos globais em $\mathbf{f}$, como rotação e translação, como movimentos locais de objetos na cena.

Em relação à reconstrução de imagens únicas, a super-resolução emprega um termo *data fidelity* estendido, onde a imagem estimada deve estar de acordo não apenas com uma observação, mas com várias observações. Se os deslocamentos entre as observações (p. ex. translações) forem em números inteiros de *pixels*, a informação

contida em diferentes quadros é a mesma, pois isso equivale a amostrar a cena nas mesmas posições com atraso. Por outro lado, se os deslocamentos contiveram frações de *pixels*, há de fato informação adicional e o número de possíveis soluções ao problema é reduzido, portanto, tornando o problema mais bem-posto.

De forma genérica, a solução é dada por

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \ \lambda\varphi(\mathbf{f}) + \sum_{k} \|\mathbf{g}_k - \mathbf{RHM}_k\mathbf{f}\|^2 . \tag{4.10}$$

Ainda no âmbito dos estudos preliminares de modelos PAR, propôs-se uma extensão do algoritmo SAI para realizar super-resolução. A motivação foi usar um mesmo *framework* para considerar regularidades geométricas na imagem (como fazem os algoritmos de interpolação) e informação adicional presentes em outros quadros de um vídeo, que algoritmos tradicionais de super-resolução obtêm através da matriz de movimentos $\mathbf{M}_k$.

Conduz-se o leitor, neste momento, ao Apêndice C, página 73, onde os estudos e desenvolvimentos preliminares são descritos. Opcionalmente, o Apêndice G fornece uma revisão bibliográfica e mais detalhes sobre métodos de super-resolução.

Capítulo 5

Modelos revisitados

Os experimentos iniciais descritos no Apêndice C demonstraram o potencial do uso de modelos autorregressivos por partes (PAR) como um regularização localmente adaptativa para problemas inversos em imagens. O passo seguinte, e natural, seria o uso de modelos PAR como regularização em problemas de super-resolução explorando-se, por exemplo, a seguinte solução

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \ \lambda \left\| (\mathbf{I} - \mathbf{A})\, \mathbf{f} \right\|_{\phi}^{\phi} + \sum_{k} \left\| \mathbf{g}_k - \mathbf{RHM}_k \mathbf{f} \right\|^2 . \tag{5.1}$$

Entretanto, antes de avançar, sentiu-se a necessidade de um estudo mais aprofundado sobre *priors* e estatísticas de imagens naturais. De maneira geral, a qualidade da solução em problemas inversos depende fortemente da capacidade do modelo descrever bem a resposta desejada. Neste capítulo serão abordadas algumas questões envolvendo modelos, de modo a prover uma base mais sólida para desenvolvimentos futuros.

5.1 Experimento com modelos PAR

Inicialmente, foi realizado um experimento com o intuito de verificar o desempenho de diferentes estratégias de estimação na qualidade de modelos PAR. A influência no resíduo de diversas variáveis, como por exemplo, ordem do modelo, tamanho da janela de treinamento, uso de mínimos quadrados ponderados (*weighted least-squares*), entre outros, foi analisada.

As métricas foram calculadas a partir do resíduo

$$\mathbf{e} = \mathbf{f} - \mathbf{A}_{\mathbf{g}}\mathbf{f}, \tag{5.2}$$

onde a matriz do modelo $\mathbf{A}_{\mathbf{g}}$ é estimada a partir de uma observação degradada (com ruído e *blurring*) $\mathbf{g}$ de $\mathbf{f}$.

Para cada combinação de variáveis, foram calculados erro médio, erro absoluto máximo e formato da distribuição do resíduo para um grupo de imagens. Além disso, foi analisada a variância dessas medidas, ou seja, sua consistência entre diferentes imagens.

5.2 Estudo sobre *priors*

Uma pergunta pertinente é se modelos PAR são realmente boas opções como *priors* em problemas inversos de processamento de imagens. Para tentar responder, mesmo que parcialmente essa pergunta, foram analisadas algumas metodologias de modelagem de imagens usadas na literatura. O enfoque foi no uso dos modelos em problemas inversos e as metodologias consideradas relacionadas aos modelos PAR são descritas com mais detalhes. Conduz-se o leitor, neste momento, aos Apêndices D e E, iniciando na página 118, onde detalhes sobre os experimentos e estudos sobre *priors* são desenvolvidos.

Capítulo 6

Trabalhos futuros e novas direções

Após os experimentos e estudos sobre novos algoritmos e tendências no campo de problemas inversos, serão delineados, neste capítulo, trabalhos futuros e novas direções de pesquisa.

6.1 Análise versus síntese

No que tange análise versus síntese, apesar do boom inicial no uso de representações esparsas em problemas inversos de imagens, novas descobertas vêm apontando para a superioridade da abordagem por análise sobre a por síntese [68–76]. Neste sentido, acredita-se que o uso de modelos autorregressivos por partes (PAR) e da abordagem via análise utilizada neste trabalho estão alinhados com essas descobertas, justificando, então, que se mantenha a linha de pesquisa.

6.2 Potencial apropriado

Como visto na Seção 5.1 e Apêndice D, o resíduo $\mathbf{e} = \mathbf{f} - \mathbf{A_g f}$ tende a ter parâmetro de formato $p = 1$ quando aproximado por uma distribuição gaussiana generalizada.

De fato, simulações (não descritas nesta tese) mostraram melhoras substituindo a norma ℓ_2 por ℓ_1 nos algoritmos dos experimentos iniciais apresentados no Capítulo 4 e Apêndice C, ou seja, utilizando-se

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \ \|\mathbf{W}\mathbf{f} - \mathbf{g}\|^2 + \lambda \|(\mathbf{I} - \mathbf{A})\,\mathbf{f}\|_1 \,. \tag{6.1}$$

Embora o uso da norma ℓ_1 no lugar de ℓ_2 dificulte a solução (que é em geral lenta por métodos tradicionais de minimização ℓ_1-ℓ_2 [77]), novas estratégias via algoritmos *iterative-shrinkage*, como o método *augmented Lagrangian* apresentado no Capítulo 3 e Apêndice B, podem ser usadas para acelerar os cálculos.

O método *augmented Lagrangian* não poderia ser aplicado diretamente, pois a matriz $\mathbf{A}$ não é circulante por blocos e não permitiria solução por FFT. Uma opção seria usar o método BiCGstab(ℓ) [78] no sub-problema que envolve a matriz $\mathbf{A}$ ou uma abordagem por *forward-backward splitting*, que é semelhante porém não requer inversão, somente gradientes [52, 79] (neste caso, multiplicações por $\mathbf{A}^{\mathrm{T}}$).

6.2.1 Classes e potenciais

Métodos como FoE (*Field of Experts* [80]) permitem que a importância do potencial, representado pelo parâmetro α, seja aprendido e controlado para cada classe.

Porém, FoE é genérico para qualquer imagens e o seu uso é "*off-line*". Primeiramente, os parâmetros são treinados para um conjunto grande de *patches* e em seguida, sem iteração entre os processos, as imagens são reconstruídas usando o modelo.

No caso de classes que são iterativamente adaptadas à imagem sendo reconstruída (BM3D [81], PLE [66]), pode-se argumentar que a iteração EM (*expectation-minimization*) produz classes com importâncias equivalentes, e portanto o parâmetro α se torna desnecessário.

6.3 Treinamento de modelos PAR orientado a classes

Ao se atribuir um modelo diferente para cada *pixel* de uma imagem (como foi feito nos experimentos apresentados), teve-se a intenção de permitir o máximo grau de liberdade para o modelo. Porém, a literatura tem indicado que abordagens não locais (*nonlocal*), que exploram repetições de estruturas na imagem, obtêm resultados melhores.

A estratégia seria fixar um número de classes muito menor que o número de *pixels* da imagem ($C \ll MN$), sendo que cada *patch* pertence a somente uma classe. Pelos algoritmos revistos, o número de classes ideal varia entre $C = 15, \cdots, 18$. Isso apresenta algumas vantagens.

A primeira seria que o número de *patches* disponíveis para a estimação dos parâmetros do modelo PAR seria maior que somente os *patches* semelhantes da vizinhança. Isso tornaria a estimativa mais estabilizada e permitiria aumentar a ordem do modelo de modo a capturar estruturas mais complexas.

O problema de *overfitting*, causado por modelos de ordens maiores, seria atenuado, pois os modelos seriam menos específicos mesmo que adaptados às características locais da imagem. O uso de classes também captura uma dependência entre diferentes

locais da imagem (*nonlocal*), pois as mesmas são treinadas e adaptadas ao conteúdo da imagem, implementando o dito *collaborative filtering*.

Em [57], essas ideias foram usadas para resolver um problema de *compressive sensing*. Experimentos demonstraram que os resultados usando classes foram consistentemente superiores. Como argumentado, embora as estatísticas de segunda ordem variem ao longo das imagens (ou seja, imagens são estacionárias apenas localmente), essa mudança pode ser periódica de forma que o sinal se repita em diferentes locais da imagem.

6.4 Remoção da componente DC

Dois *patches* distintos na imagem podem ter as mesmas características, diferenciando apenas pelo valor DC. Isso pode ser considerado comum, já que uma possível causa são variações de iluminação da cena. Tais *patches* poderiam, portanto, ser considerados da mesma classe, enriquecendo, assim, o conteúdo da mesma e estabilizando a estimação do modelo.

Para que o mesmo modelo sirva para *patches* com componentes DC distintas, uma restrição adicional deve ser imposta: que a soma dos coeficientes do modelo seja 1, pois

$$f = \sum_i a_i f_i \tag{6.2}$$

$$f + \mathrm{DC} = \sum_i a_i (f_i + \mathrm{DC}) = f + \mathrm{DC} \sum_i a_i \Rightarrow \sum_i a_i = 1. \tag{6.3}$$

Desta maneira, uma equação extra é acrescentada e os graus de liberdade são reduzidos. Isso pode ter efeitos benéficos sobre a estimação do coeficientes. Em [13] há uma discussão sobre os efeito da remoção do DC na estatística de imagens naturais. Acredita-se que seja interessante alguma investigação nesse sentido.

6.5 Expectation-Maximization

Dependendo do nível de ruído, borramento e subamostragem de imagem, a estimação dos parâmetros do modelo pode se tornar difícil (ver Seção 5.1 e Apêndice D). Uma estratégia que pode ser empregada, e que foi usada com sucesso em [66], é *expectation-minimization*. Trata-se de um processo iterativo que alterna entre estimação da resposta (passo E - *expectation*) e estimação dos parâmetros de um modelo que descreva a resposta (passo M - *maximization*). Nas primeiras iterações é possível que a qualidade dos parâmetros do modelo não permita ao passo E chegar à resposta

desejada. Porém, se o modelo for suficiente para aproximar a resposta ótima, o passo M reestimará o modelo, melhorando sua qualidade para as próximas iterações.

Um problema visível de tal abordagem é a convergência, que depende fortemente da inicialização. Caso o modelo inicial esteja muito longe do ideal, é possível que a sequência dos passos não venha a convergir. Uma análise de convergência, assim como exemplos que não convergem, podem ser encontradas em [82].

Em [66] foi observado que, para os casos de interpolação, a inicialização do modelo é de suma importância para convergência. Experimentos indicaram que as bases PCA iniciais que levam à convergência são aquelas que descrevem bem contornos. Foram usadas $C = 18$ classes e cada uma representava um ângulo de um contorno. Nos 180° de ângulos possíveis, isso significa uma discretização de 10°. Contornos são também o foco de outros algoritmos de interpolação do estado da arte [56, 60, 83].

6.6 Modelos para *pixels* faltantes

Em problemas de interpolação e super-resolução, quando um *prior* genérico é utilizado, como por exemplo TV, todos os *pixels* (os existentes e os que serão estimados) já têm *a priori* o mesmo modelo que os descreva. Neste trabalho, porém, a cada *pixel* é atribuído um modelo distinto que depende da sua vizinhança. É preciso, portanto, um procedimento para atribuir modelos aos *pixels* não existentes na imagem observada.

Nos experimentos aprestados, a heurística usada foi atribuir aos *pixels* faltantes um modelo cujos parâmetros são calculados a partir da média dos parâmetros dos modelos dos *pixels* vizinhos. Contudo, reconhecidamente, essa estratégia tem falhas. A mais clara acontece em quinas, onde dois *pixels* vizinhos são descritos por dois modelos totalmente diferentes.

Na classe de algoritmos que contém o algoritmo SAI de [59], é assumido que os modelos estimados em LR se aplicam aos *pixels* em HR. Como isso nem sempre é válido, algumas estratégias heurísticas são usadas para tornar o processo robusto. Em [60], por exemplo, mínimos quadrados ponderados são adicionalmente usados para melhorar a robustez.

Outros métodos partem de uma imagem piloto que é estimada com um algoritmo tradicional, por exemplo interpolação bicúbica, para em seguida calcular e atribuir os modelos a todos os *pixels* [57, 84, 85].

Acredita-se que uma abordagem mais adequada a ser seguida é aquela usada em [66]. Os modelos iniciais para qualquer *pixel* são aqueles que descrevem bem contornos em diversos ângulos. Em seguida, pelo algoritmo EM, os modelos são iterativamente adaptados ao conteúdo da imagem.

6.7 Seleção do parâmetro de regularização

Na literatura em geral, o parâmetro de regularização λ que controla a importância entre *prior* e fidelidade às observações é selecionado empiricamente e de forma manual. Basicamente, testa-se o algoritmo para diversos valores de λ e seleciona-se aquele que fornece a melhor qualidade de reconstrução. Contudo, tal abordagem não é prática, pois em casos reais não se tem acesso à imagem original.

Intuitivamente, o valor de λ depende

- do nível de ruído da observação, pois quanto mais ruidosa a imagem, mais importância deve ser dada ao *prior*;
- do teor de degradação ou *blurring*, com a mesma filosofia;
- da qualidade do *prior*, pois se ele descreve bem a imagem, deve receber mais peso;
- da energia da imagem quando comparada à energia do ruído.

O método mais utilizado na literatura é através da curva-L ou *L-curve* [86]. Um método relacionado é através da *U-curve* [87], que soluciona algumas limitações da *L-curve*. Outros métodos que abordam seleção de λ podem ser encontradas em [38, 79, 88, 89].

Como aparentemente não há nenhum trabalho que ataque esse problema do ponto de vista de *priors* por modelos PAR, acredita-se que este seja um campo onde há espaço para pesquisa.

6.8 Super-resolução

Inspirando-se nos desenvolvimentos apresentados no Apêndice C.4 e nos trabalhos [88–90], pretende-se estudar a seguinte abordagem para super-resolução simultânea

$$\hat{\mathbf{f}}_1, \cdots, \hat{\mathbf{f}}_K = \underset{\mathbf{f}_1,\cdots,\mathbf{f}_K}{\arg\min} \sum_{k=1}^{K} \|\mathbf{R}_k\mathbf{H}_k\mathbf{f}_k - \mathbf{g}_k\|^2 + \sum_{k=1}^{K-1}\sum_{l=0}^{L} \|\Lambda_{k,k+l}(\mathbf{f}_k - \mathbf{A}_{k,k+l}\mathbf{f}_{k+l})\|_1 \tag{6.4}$$

onde

$$\mathbf{A}_{k,k+l} \rightarrow \begin{cases} l \neq 0, & \text{Matriz } \mathbf{A} \text{ modela movimento entre quadros} \\ l = 0, & \text{Matriz } \mathbf{A} \text{ modela regularidade dentro do quadro} \end{cases} \tag{6.5}$$

Em suma, pretende-se num mesmo *framework* levar em conta o movimento relativo entre os quadros e *priors* para imagens individuais.

Considerem-se as situações possíveis em um vídeo:

1. Movimento entre quadros bem definido e facilmente estimável: é possível aplicar estimação por super-resolução;

2. Quadros consecutivos totalmente diferentes, como numa mudança de cena: não é possível super-resolução, porém é possível interpolação e *deblurring* usando *priors* adequados;

3. Situações combinadas, por exemplo com movimentos locais.

Para levar em conta todos esses casos, pode-se usar uma matriz de pesos $\Lambda_{k,k+l}$ controla a importância entre a informação de movimento relativo entre quadros e regularidade dentro da imagem. A matriz de pesos pode ser gerada a partir da confiança que se tem na estimação de movimento.

Neste caso, $\Lambda_{k,k+l}$ seria uma matriz diagonal e

$$\sum_{l=0}^{L} \Lambda_{k,k+l} = \mathbf{I}. \tag{6.6}$$

Em algoritmos de super-resolução, movimentos entre quadros envolvendo números inteiros de *pixels* são considerados sem valia, pois não provêm uma visão diferente da cena e, portanto, nenhuma informação extra. Por outro lado, tais movimentos inteiros fornecem mais *patches* e, portanto, enriqueceriam as classes, melhorando a estimação dos modelos para *priors*.

Como em [90], pretende-se estudar uma abordagem simultânea, onde todos os quadros são estimados de maneira paralela. Em notação vectorial, fazendo

$$\mathbf{f} = [\mathbf{f}_1^{\mathrm{T}}, \cdots, \mathbf{f}_K^{\mathrm{T}}]^{\mathrm{T}} \tag{6.7}$$

$$\mathbf{g} = [\mathbf{g}_1^{\mathrm{T}}, \cdots, \mathbf{g}_K^{\mathrm{T}}]^{\mathrm{T}} \tag{6.8}$$

$$\mathbf{R} = \mathrm{diag}(\mathbf{R}_1, \cdots, \mathbf{R}_K) \tag{6.9}$$

$$\mathbf{H} = \mathrm{diag}(\mathbf{H}_1, \cdots, \mathbf{H}_K) \tag{6.10}$$

pode-se reescrever (6.4) como

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \ \|\mathbf{RHf} - \mathbf{g}\|^2 + \|\Lambda\left(\mathbf{I} - \mathbf{A}\right)\mathbf{f}\|_1 . \tag{6.11}$$

Uma opção de simplificação é considerar apenas os quadros dois a dois e, fazendo $L = 1$, tem-se

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{2,2} & \mathbf{A}_{2,3} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{A}_{K-1,K-1} & \mathbf{A}_{K-1,K} \end{bmatrix} \tag{6.12}$$

e

$$\Lambda = \begin{bmatrix} \Lambda_{1,1} & \Lambda_{1,2} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Lambda_{2,2} & \Lambda_{2,3} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \Lambda_{K-1,K-1} & \Lambda_{K-1,K} \end{bmatrix} \tag{6.13}$$

6.8.1 Expectation-Minimization

Pode ser interessante iterar algumas vezes entre estimação do modelo e estimação da resposta seguindo a filosofia EM. Matematicamente, ter-se-ia

$$\hat{\mathbf{f}}_{q+1} = \arg\min_{\mathbf{f}} \ \|\mathbf{RHf} - \mathbf{g}\|^2 + \|\Lambda_q(\mathbf{f} - \mathbf{A}_q\mathbf{f})\|_1 \tag{6.14}$$

$$\hat{\mathbf{A}}_{q+1} = \arg\min_{\mathbf{A}} \ \|\mathbf{RHf}_{q+1} - \mathbf{g}\|^2 + \|\Lambda_q(\mathbf{f}_{q+1} - \mathbf{Af}_{q+1})\|_1 \tag{6.15}$$

$$\hat{\Lambda}_{q+1} = \arg\min_{\Lambda} \ \|\mathbf{RHf}_{q+1} - \mathbf{g}\|^2 + \|\Lambda(\mathbf{f}_{q+1} - \mathbf{A}_{q+1}\mathbf{f}_{q+1})\|_1 \tag{6.16}$$

6.9 Conclusão

Este capítulo apresentou delineamentos de trabalhos futuros envolvendo tanto mudanças na estimação dos modelos PAR quanto o uso deles em esquemas de super-resolução. Tais mudanças foram inspiradas em trabalhos recentes e novas tendências na área de problemas inversos em processamento de imagens.

Capítulo 7

Correção de ruído de padrão fixo

Neste capítulo, outro tipo de ruído é abordado, o dito ruído de padrão fixo (FPN - *fixed pattern noise*), muito comum em vídeos na faixa do infravermelho. Por ter características peculiares, a remoção de FPN requer diferentes abordagens, que são apresentadas a seguir.

7.1 Introdução

Na atualidade, câmeras infravermelhas e vídeos na faixa do infravermelho são usados em inúmeras áreas como ensaios não-destrutivos para verificação da integridade de equipamentos, visão noturna, segurança, reconhecimento e vigilância aeroespacial, imagens térmicas astronômicas e aplicações militares [91].

No que tange aos sensores, as últimas décadas presenciaram o aparecimento de dispositivos cada vez mais precisos e baratos. As FPA's, ou *focal plane arrays*, são sensores de imagem que consistem de uma matriz de sensores ópticos localizada no plano focal de um sistema de lentes. A sua aparição possibilitou a construção de dispositivos para aquisição de imagens e vídeos que são registrados diretamente em formato digital.

Na faixa do infravermelho, as IRFPA's (*infrared focal plane arrays*) têm se tornado o mais proeminente detector usado em sistemas de imagens nos últimos anos. Seu vasto uso é atribuído aos avanços na tecnologia de sensores de estado sólido, que permitiram compacidade, baixo custo e alto desempenho.

Sabe-se que um problema comum a todos os sensores IRFPA é o ruído de padrão fixo (*fixed-pattern noise* ou FPN), também chamado de não-uniformidade espacial. De fato, o FPN continua sendo um sério problema, apesar dos avanços recentes nessa tecnologia. A origem deste ruído é atribuída ao fato de cada detector da matriz, ou seja, cada *pixel* possuir uma variação no processo de fabricação. Em outras palavras, cada *pixel* do detector responde de maneira diferente à mesma quantidade de radiação incidente. O ruído FPN se manifesta aleatoriamente no espaço e está

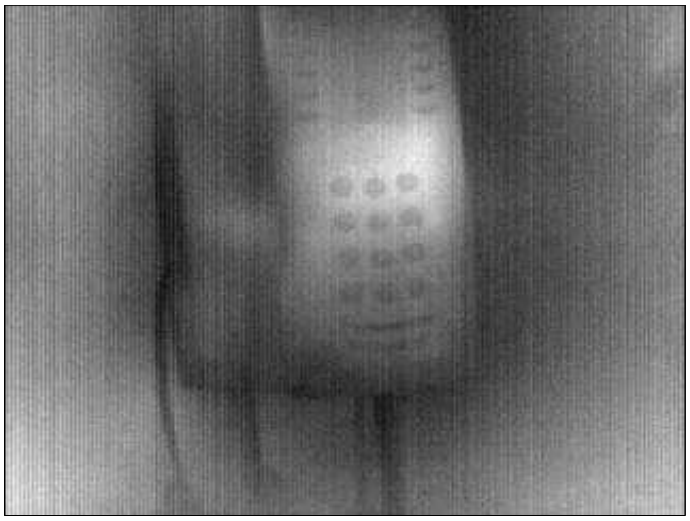


Figura 7.1: Vídeo real com ruído de padrão fixo.

presente em todos os quadros de um vídeo infravermelho independentemente da cena ou movimento [91]. A Figura 7.1 mostra um vídeo real contaminado com ruído de padrão fixo. Nota-se que o FPN degrada consideravelmente a qualidade da imagem.

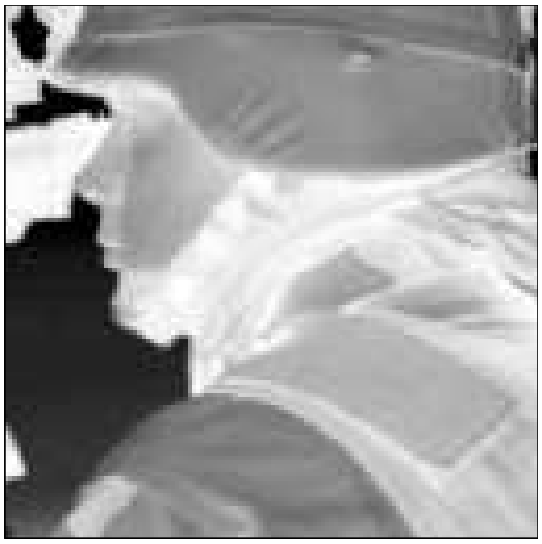
Outra característica do ruído FPN é sua lenta variação temporal durante o funcionamento do sensor. Ele pode se tornar significativo numa ordem de grandeza de 30 segundos após uma calibração [92]. Esse desvio temporal é atribuído a variações na temperatura do sensor, material de fabricação, ruído eletrônico de leitura, controle automático de ganho, entre outros. Portanto, uma única calibração é ineficaz e o problema requer estimativa e compensação contínuas durante a operação da câmera [91].

Embora a verdadeira resposta das IRFPA's seja não-linear, ela é em geral modelada linearmente como função da radiância, um ganho e um desvio (*bias*) [91, 93] como

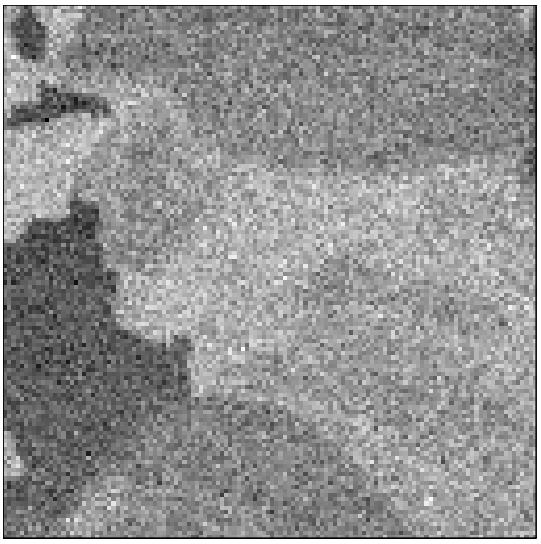
$$g_k(n_1, n_2) = a(n_1, n_2) f_k(n_1, n_2) + b(n_1, n_2), \tag{7.1}$$

onde $g_k(n_1, n_2)$ é a saída da câmera referente ao *pixel* (n_1, n_2) no instante k, $a(n_1, n_2)$ é o ganho associado ao *pixel* (n_1, n_2), $f_k(n_1, n_2)$ é a radiância incidente no elemento sensor (n_1, n_2) no instante k e $b(n_1, n_2)$ é desvio ou *bias* associado ao *pixel* (n_1, n_2).

A Figura 7.2a ilustra um quadro de um vídeo infravermelho sintético onde não existe FPN. A Figura 7.2b mostra o mesmo quadro contaminado com FPN segundo o modelo da equação (7.1).



(a) Quadro original de um vídeo infravermelho.



(b) Quadro contaminado com ruído de padrão fixo.

Figura 7.2: Exemplo de ruído de padrão fixo sintético.

Dando prosseguimento ao trabalho realizado no mestrado do candidato [94], foi desenvolvido durante o período do doutorado o algoritmo *Affine Projection* para correção de ruído de padrão em vídeos infravermelhos. A partir desse desenvolvimento, foi submetido e aceito um artigo [2] em periódico internacional. Conduz-se o leitor, neste momento, ao Apêndice F, página 145, onde os desenvolvimentos são apresentados.

Capítulo 8

Conclusão e sumário

Esta tese apresentou um estudo sobre algumas técnicas para resolver problemas inversos em processamento de imagem e vídeo. As contribuições geradas foram resumidas na Seção 1.3.

Problemas inversos

Nos Capítulos 1 e 2 o conceito de problema inverso em processamento de imagens foi introduzido. Foi visto que métodos tradicionais, como interpolação bicúbica e filtro de Wiener, não geram boas respostas, pois não modelam bem imagens naturais e, portanto, não fornecem *priors* adequados para suprir a falta de informação perdida durante o processo de degradação.

Total variation

Como um primeiro passo para uma melhor modelagem de imagens naturais, foi estuda a técnica de deconvolução por *total variation* no Capítulo 3. Foi ilustrado como as estatísticas de imagens naturais justificam o uso de TV. Mais especificamente, embora a estatística de *pixels* individuais não seja consistente entre diferentes imagens, a estatística das derivadas horizontal e vertical é mais bem comportada. O fato de tais estatísticas terem distribuições Laplacianas justifica o uso da norma ℓ_1. Outra interpretação é que o *prior* TV promove respostas suaves por partes, fato observado em imagens naturais. Aliada a isso, a existência de métodos rápidos torna a abordagem por TV bastante atrativa.

Foi apresentada a abordagem por *augmented Lagrangian*, um método particularmente eficiente para resolver problemas envolvendo TV. Basicamente, a parte não diferenciável é separada (via *variable splitting*) e atacada por métodos de *shrinkage*. Por combinar penalização quadrática com multiplicadores de Lagrange, que são explicitamente estimados durante o processo, a abordagem por *augmented Lagrangian* atinge uma ótima taxa de convergência quando comparada a outras abordagens.

Uma modificação desse método foi proposta no Apêndice B, onde o operador de gradiente foi substituído por filtros direcionais seguidos de derivadas direcionais. A motivação foi de selecionar regiões de uma imagem mais apropriadas para um determinado tipo de penalização antes de aplicá-la. O algoritmo proposto produz resultados superiores ao TV tradicional em termos de PSNR, além de possuir outras vantagens, como fornecer maior PSNR para uma faixa mais ampla de coeficientes de regularização. Esses desenvolvimentos tiveram publicação aceita em anais de congresso internacional [3].

Modelos PAR

Sem uma preocupação inicial com algoritmos rápidos, como foi o caso de TV, mas com enfoque em uma melhor descrição de imagens, começou-se a estudar a partir do Capítulo 4 regularizações localmente adaptativas. Em particular, foram estudadas aplicações de modelos autorregressivos por partes (PAR) em interpolação de imagens. O algoritmo SAI [59] foi estendido de modo a considerar a imagem como um todo em vez de dividi-la em blocos. A partir disso, o *prior* foi generalizado para $\|(\mathbf{I}-\mathbf{A})\,\mathbf{f}\|_p^p$, que fornece modelos individuais para cada *pixel* da imagem. Vale mencionar que trabalhos posteriores também começaram a usar esse formato de *prior* [56, 57].

No Apêndice C, estudos preliminares mostraram ser possível melhorar a PSNR da interpolação de imagens usando o *prior* generalizado. O caso de imagens pré-filtradas antes da subamostragem também foi considerado de maneira acoplada e experimentos mostraram melhoras sutis, porém consistentes, em relação a abordagens desacopladas (interpolação seguida de *deblurring*).

Outra extensão do algoritmo SAI, agora para efetuar super-resolução, também foi proposta no Apêndice C. O método foi inspirado em algoritmos que não usam estimação de movimento explicitamente, como [84] e [85]. No esquema proposto, tanto regularidades geométricas quanto movimento relativo entre quadros são capturados através dos modelos PAR. Em seguida, a confiança em cada modelo, medida pelo MSE, é usada para combinar as estimativas.

Embora acredite-se que os resultados inicias tenham sido encorajadores, algumas questões relacionadas à ordem do modelo a ser usada, estratégias de estimação do modelo e possível inferioridade da abordagem via modelos PAR a outras, conduziram os estudos a uma reanálise da literatura de problemas inversos.

Modelos revisitados

Antes de partir diretamente para o uso do *prior* $\|(\mathbf{I}-\mathbf{A})\,\mathbf{f}\|_p^p$ para realizar super-resolução, escolheu-se dar uma passo atrás: um estudo mais aprofundado sobre estatísticas de imagens naturais e *priors* foi realizado e apresentado no Capítulo 5.

A ideia foi revisitar a abordagem via modelos PAR à luz de novos desenvolvimentos e descobertas na área de problemas inversos. A intenção inicial do estudo foi reunir informações para sustentar uma escolha: manter ou abandonar a abordagem via modelos PAR, com possível adoção em trabalhos futuros de técnicas como representações esparsas.

Inicialmente, no Apêndice D, foi realizado um experimento para avaliar o efeito de diversas variáveis (como tamanho do modelo, janela de treinamento) na qualidade de representação dos modelos PAR. Várias combinações de tais variáveis foram usadas para estimar os parâmetros do modelo a partir de imagens corrompidas com diversos níveis de ruído e *blurring*.

As conclusões do experimentos indicaram, por exemplo,

- que o resíduo tem distribuição gaussiana generalizada com parâmetro de forma $\phi \approx 1.1$, portanto mais próxima da Laplaciana;
- que o treinamento do modelo usando mínimos quadrados ponderados (WLS - *weighted least squares*), dando mais peso a *patches* com vizinhança semelhante, é mais robusto;
- que modelos de ordens maiores tem desempenho melhor, provavelmente pela flexibilidade fornecida por um maior número de coeficientes;

Embora os resultados obtidos nos experimentos tenham melhorado a intuição sobre os modelos PAR, pontos importantes não foram investigados. Por exemplo, como mencionado na literatura [56, 57, 59], modelos de ordem maior tendem a causar *overfitting* e gerar respostas com oscilações não desejadas, e tal fato não foi avaliado.

No Apêndice E, vários modelos para imagens naturais e abordagens de problemas inversos foram apresentados. Inicialmente, as abordagens por análise e síntese, tipicamente usadas em problemas inversos, foram revisadas, apontando-se suas semelhanças e diferenças. Salientou-se o resultado de [68], que apontou que para o caso *overcomplete* (síntese via dicionários redundantes ou análise com "muitos" analisadores) as abordagens deixam de ser equivalentes.

Modelos generativos, que têm uso tradicionalmente na abordagem por síntese, foram recapitulados. A compactação de energia da DCT e esparsidade das transformadas Wavelet e descendentes foram relembradas com o enfoque no seu uso como *priors*. A obtenção de dicionários a partir de bancos de dados de imagens naturais também foi analisada. Algoritmos para treinamento de dicionários redundantes, como o K-SVD [95], foram rapidamente mencionados.

Modelos baseados em energia, especialmente os inspirados em campos aleatórios de Markov (MRF - *Markov random fields*), foram revisados na Seção E.3.1. Atenção especial foi dada à técnica *Field of Experts* (FoE), que fornece um dos melhores resultados de reconstrução de imagem das técnicas baseadas em MRF. FoE, de certa

forma, generaliza a abordagem por TV, substituindo a norma ℓ_1 por potenciais *Student-t* e o gradiente $\mathbf{D}$ por filtros. Tanto os parâmetros dos potenciais quanto os filtros, são treinados a partir de um conjunto amplo de imagens naturais.

Em seguida, três métodos, BM3D [81], PLOW [96] e PLE [66] foram apresentados. São técnicas que exploram a repetição de estruturas dentro da imagem e são consideradas estado-da-arte em termos de *denoising*, interpolação e *deblurring*. Esses métodos superam os anteriores, como FoE, que pretende ser um modelo genérico para imagens, treinado a partir de um banco de dados, porém não adaptado particularmente à imagem que está sendo reconstruída.

As características marcantes dessas três técnicas são:

- Abordagem por análise;
- Paradigma não nocal (*nonlocal*): supõe que, embora somente localmente estacionárias, estatísticas de segunda ordem podem se repetir dentro da imagem;
- *Collaborative filtering*: todos os *patches* identificados como semalhantes, serão usados para estimar o filtro que os filtrará.
- *Structured sparsity*: assume que a imagem é esparsa em um dicionário redundante, porém a seleção dos átomos durante a reconstrução é restrita a um subconjunto pré-selecionado do dicionário, tornando a reconstrução mais estável.

Ilustrou-se a semelhança da abordagem por modelos PAR a outros algoritmos do estado-da-arte, como PLE, em termos de função custo. Embora tal fato indique o mantenimento da abordagem, a parte crítica do problema, que envolve a estimação dos modelos, deve ser revista em trabalhos futuros.

Finalmente, foi discutido intuitivamente por que a abordagem via análise é em geral superior à síntese. Tal fato, inicialmente observado e provado em [68], tem recebido maior atenção e vem sendo corroborado em trabalhos recentes [69–76].

Trabalhos futuros e novas direções

Uma vez posicionados em relação a outros *priors*, foram levantadas possíveis modificações nas estratégias de estimação de modelos PAR usadas no Apêndice C e D. Entre elas estão: classificação de *patches* com estimação de modelos dentro das classes e iterações entre cálculo dos parâmetros do modelo e cálculo da resposta (algoritmo EM).

Na Seção 6.8, foram propostos esquemas de super-resolução usando modelos PAR inspirados nos desenvolvimentos da Seção 5 e em [90]. Trabalhos futuros incluem implementação dessas ideias e avaliação de desempenho dos algoritmos.

Ruído de padrão fixo

Outro tipo de reconstrução de vídeo, agora corrompido com ruído de padrão fixo (FPN), foi considerada no Capítulo 7. Foi desenvolvida, durante o doutorado, uma extensão dos métodos de remoção FPN em vídeos infravermelhos inicialmente desenvolvidos em [4, 97]. A partir de modificações no algoritmo RLS, foi elaborado um esquema AP (*affine projection*) que resultou em uma convergência mais rápida e menor erro de estado estacionário quando comparado à abordagem RLS. Tais desenvolvimentos, apresentados no Apêndice F, tiveram publicação aceita em periódico internacional [2].

Apêndice A

Publicações

Segue abaixo a lista de publicações até a presente data do candidato ao doutorado.

A.1 Artigos em periódicos internacionais

Total de artigos aceitos em periódicos internacionais: 3.

1. PIPA, D. R., DA SILVA, E. A. B., PAGLIARI, C. L., DINIZ, P. S. R. "Recursive Algorithms for Bias and Gain Nonuniformity Correction in Infrared Videos", Aceito para publicação em 28/08/2012 na revista *IEEE Transactions on Image Processing*, `http://dx.doi.org/10.1109/TIP.2012.2218820`, Fator de Impacto: 3.042, Classificação Qualis: A1, ISSN: 1057-7149, [2].

2. OKAMOTO, J., GRASSI, V., AMARAL, P., PIPA, D. R., et al. "Development of an Autonomous Robot for Gas Storage Spheres Inspection", *Journal of Intelligent and Robotic Systems*, 2012, `http://dx.doi.org/10.1007/s10846-011-9607-z`, Fator de Impacto: 0.829, Classificação Qualis: A2, ISSN: 0921-0296, [98].

3. PIPA, D., MORIKAWA, S., PIRES, G., et al. "Flexible Riser Monitoring Using Hybrid Magnetic/Optical Strain Gage Techniques through RLS Adaptive Filtering", *EURASIP Journal on Advances in Signal Processing*, 2010, `http://dx.doi.org/10.1155/2010/176203`, Fator de Impacto: 1.012, Classificação Qualis: A1, ISSN: 1687-6172, [99].

A.2 Artigos em congressos internacionais

Total de artigos publicados em anais de congressos internacionais: 5.

1. PIPA, D., CHAN, S. H., NGUYEN, T. "Directional Decomposition Based Total Variation Image Restoration". In: *20th European Signal Processing Conference*, 2012, [3].

2. OKAMOTO, J., GRASSI, V., AMARAL, P. F. S., PIPA, D. R., et al. "Autonomous Robot for Welding Line Following for Gas Storage Spheres Inspection". In: *Mechatronics2010 - The 12th Mechatronics Forum Biennial International Conference*, 2010, [100].

3. PIPA, D. R., DA SILVA, E. A. B., PAGLIARI, C. L., et al. "Joint Bias and Gain Nonuniformity Correction of Infrared Videos Using Tensorial- RLS Technique". In: *Image Processing, 2009. ICIP 2009. 16th IEEE International Conference on*, 2009, [97].

4. MORIKAWA, S. R. K., CAMERINI, C. S., PIPA, D. R., et al. "Monitoring of Flexible Oil Lines Using FBG Sensors". In: *International Conference on Optical Fibre Sensors (OFS)*, 2008, [101].

5. MARINHO, M. G., CAMERINI, C. S., MORIKAWA, S. R. K., PIPA, D. R., et al. "New Techniques for Integrity Management of Flexible Riser End-fitting Connection". In: *27th International Conference on Offshore Mechanics and Arctic Engineering (OMAE2008)*, 2008, [102].

A.3 Artigos em congressos nacionais

1. PIPA, D. R., DA SILVA, E. A. B., PAGLIARI, C. L. "Correção de Não-Uniformidade em Vídeo Infravermelho por Gradiente Descendente". In: *XXVI Simpósio Brasileiro de Telecomunicações - SBrT'08*, 2008, [94].

2. PIPA, D. R. "Denoising de Sinais de Ultrassom Utilizando Wavelets". In: *XXVI Congresso Nacional de Ensaios Não Destrutivos e Inspeção (CONAEND&IEV)*, 2008, [103].

3. PIPA, D. R., PIRES, G. P. "Monitoramento da Torção em Risers Flexíveis Através de Processamento de Imagens". In: *XXVI Congresso Nacional de Ensaios Não Destrutivos e Inspeção (CONAEND&IEV)*, 2008, [104].

4. PIRES, G. P., PIPA, D. R. "Classificação de Ruídos em Emissão Acústica Utilizando Redes Neurais". In: *XXVI Congresso Nacional de Ensaios Não Destrutivos e Inspeção (CONAEND&IEV)*, 2008, [105].

Appendix B

Total variation deconvolution using directional decompositions

In this chapter, we present an extension of total variation (TV) image deconvolution technique that enhances image quality over classical TV while preserving algorithm speed. Enhancement is achieved by altering the regularization term to include *directional decompositions* before the gradient operator. Such decompositions select areas of the image with characteristics that are more suitable for a certain type of gradient than another. Speed is guaranteed by the use of the augmented Lagrangian approach as basis for the algorithm. Experimental evidence that the proposed approach improves TV deconvolution is provided, as well as an outline for a future work aiming to support and substantiate the proposed method.

This chapter is an extended version of the conference paper [3].

B.1 Introduction

Image deconvolution/restoration is a classic inverse problem that has been extensively studied in the literature. In such problems, one aims to recover a clean, sharp image from a noisy, blurred and/or degraded observation. The challenge of most inverse problems is that they are ill-posed, i.e., either the direct operator does not have an inverse, or it is nearly singular. Thus, regularization is required to deal with noise and ensure a unique solution [1].

Since its introduction in 1992 by Rudin, Osher and Fatemi [20], Total Variation (TV) regularization has been successfully applied to a variety of deconvolution-related image problems [1]. The success of TV regularization relies on a good balance between the ability to describe piecewise smooth images (without penalizing possible discontinuities) and the complexity of the resulting algorithms [14].

To go on with the idea, let $\mathbf{f}$ be a vector representing an unknown image (to be

predicted) lexicographically ordered, which is observed through the model

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\nu} \tag{B.1}$$

giving rise to a blurred and noisy image $\mathbf{g}$. The blur operator is represented by $\mathbf{H}$ and $\boldsymbol{\nu}$ is the noise term such that $\boldsymbol{\nu} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. We are interested in estimating $\mathbf{f}$ given $\mathbf{g}$ and $\mathbf{H}$.

To solve (B.1) we apply the total variation (TV) approach

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \ \frac{\mu}{2} \left\|\mathbf{H}\mathbf{f} - \mathbf{g}\right\|^2 + \left\|\mathbf{f}\right\|_{\mathrm{TV2}}, \tag{B.2}$$

where μ is the regularization parameter, $\left\|\mathbf{f}\right\|_{\mathrm{TV2}} = \sum_i \sqrt{\beta_x^2 \left[\mathbf{D}_x\mathbf{f}\right]_i^2 + \beta_y^2 \left[\mathbf{D}_y\mathbf{f}\right]_i^2}$ is the isotropic total variation norm on $\mathbf{f}$ and, by definition, we set $\left\|\mathbf{f}\right\|_{\mathrm{TV2}} \stackrel{\mathrm{def}}{=} \left\|\mathbf{D}\mathbf{f}\right\|_{\mathrm{ISO}}$, with $\mathbf{D}$ being the classical gradient operator such that

$$\mathbf{D} = \begin{bmatrix} \beta_x\mathbf{D}_x \\ \beta_y\mathbf{D}_y \end{bmatrix}, \tag{B.3}$$

with matrices $\mathbf{D}_x$ and $\mathbf{D}_y$ built to perform convolution with the kernels $d_x = [-1,\ 1]$ and $d_y = [-1,\ 1]^{\mathrm{T}}$ respectively. Here, β_x and β_y, sometimes grouped as $\boldsymbol{\beta} = [\beta_x,\ \beta_y]^{\mathrm{T}}$, are constants that control the amount of horizontal and vertical regularization, respectively. For instance, if the real image $\mathbf{f}$ is expected to have some "vertical" pattern, i.e. Figure B.1, the reconstruction process should penalize preferably vertical frequencies by choosing $\beta_y > \beta_x$.

As an example, we simulated an observation of Figure B.1 by blurring it with a 9×9 Gaussian kernel with $\sigma = 3$ and adding noise up to BSNR[1]$= 25\,\mathrm{dB}$. Then, we deblurred it using the approach on (B.2) for different choices of the regularization parameter μ. Figure B.2 shows the evolution of the PSNR for some choices of $\boldsymbol{\beta}$. The choice $\boldsymbol{\beta} = [0.5,\ 1.5]$, based on previous knowledge about the real image, forces the solution to have more horizontal than vertical frequency content, resulting in better image quality and higher PSNR.

However, when the real image does not have a preferable orientation pattern, as in Figure B.3, no substantial PSNR increment is achieved by altering $\boldsymbol{\beta}$, as shown on Figure B.4. In this chapter, we address this issue.

The rest of this chapter is organized as follows: Section B.2 reviews previous work on extending TV deconvolution for images. In Section B.3 we briefly describe the augmented Lagrangian approach with focus on the characteristics that yields efficient implementation. Section B.4 sketches the strategy to be used. The proposed

[1]Blurred Signal to Noise Ratio = 10 log (Blurred signal variance / Noise variance) [dB].

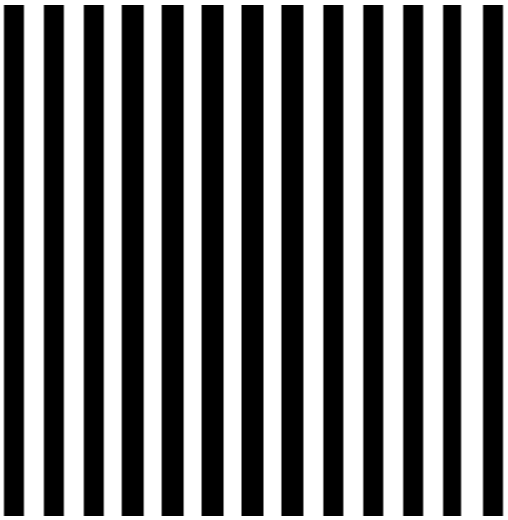


Figure B.1: A toy example showing that classical TV can be improved by choosing appropriate β_x and β_y.

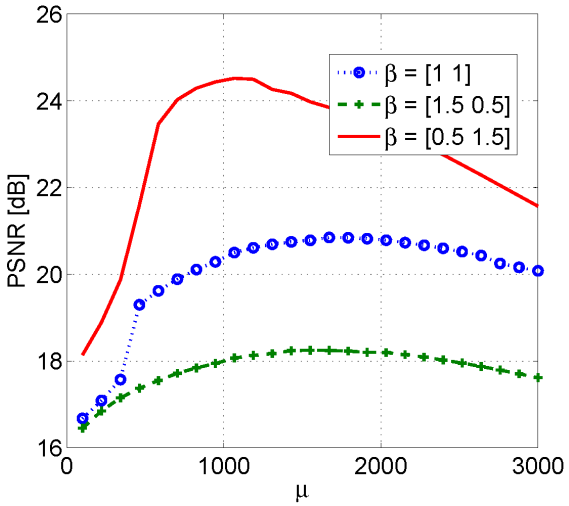


Figure B.2: PSNR is maximized when $\boldsymbol{\beta}$ is biased towards the vertical direction.

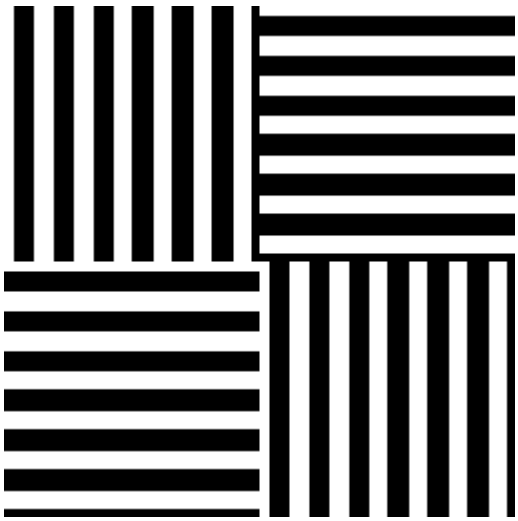


Figure B.3: A less simple toy example. Altering β_y and/or β_x does not yield better PSNR.

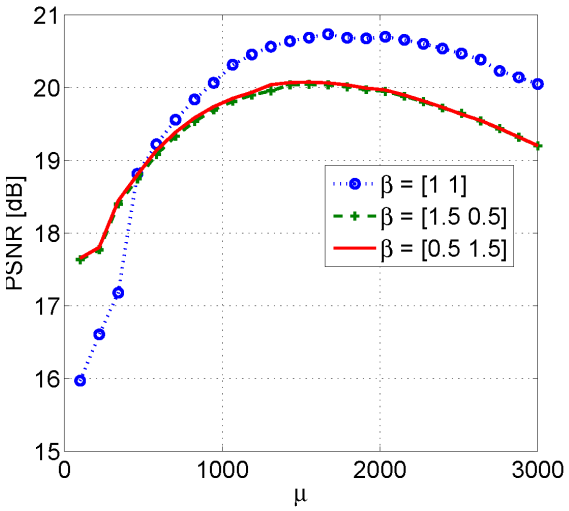


Figure B.4: Since the figure has a mixed pattern, the best result is achieve using a balanced $\boldsymbol{\beta}$.

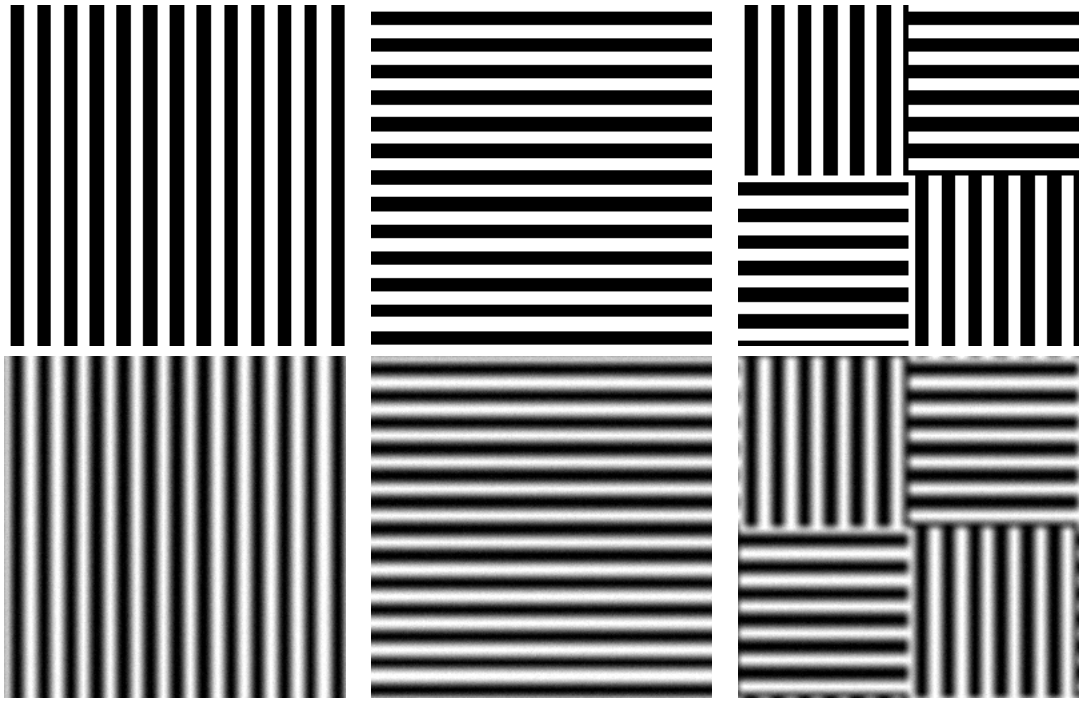


Figure B.5: Example of original (top) and blurred and noisy (bottom) images used in the experiments in Section B.1. The images were first blurred with a Gaussian kernel ($\sigma = 3$) then white Gaussian noise was added to BSNR $= 25\,\mathrm{dB}$.

method is explained in details in Section B.5 and results are provided in section B.6. Algorithmic details are presented in Section B.7 and a conclusion is drawn in Section B.9.

B.2 Related work

Many extensions of TV have been reported in the literature [29]. Most of them [26, 106, 107] deals with the *staircase effect*, namely the transformation of smooth regions (ramps) into piecewise constant regions (stairs). Such a phenomenon tends to appear when trying to reconstruct, say, a piecewise smooth image (rather than a piecewise constant image) using classical TV.

For this purpose, Chambolle and Lions proposed the use of a second order variation along with the traditional TV in [26]. In [106] Chan et al. improved the approach of [26] by considering texture and structure as separate components of an image. Stefan et al. used a variable order total variation approach in [107]. The order is chosen after an edge detection procedure.

In [108] Farsiu et al. introduced a technique called Bilateral TV, which they apply to solve a super-resolution problem [67]. Basically, rather than calculating only first-order finite differences, which is often used to approximate the gradient operator [20], they use a weighted mean of combinations of horizontal and vertical

differences. As a result, not only horizontal and vertical differences are computed, but also diagonal differences.

Kiriyama et al., in [109], proposed to speed up the Chambolle's projected method [27] by adding diagonal differences to the TV regularization term. They reported a reduction in computational time around 56% (as a result of fewer iterations).

In [110] Karahanoğlu et al. proposed the use of a general differential operator $\mathbf{L}$ instead of the derivative $\mathbf{D}$ for 1-D signal processing. Specifically, $\mathbf{L}$ can be tuned according to the expected signal and the presence of a linear system.

Differently from the previously proposed techniques, our approach uses directional filters to decompose the image into directional components. Then, we apply the appropriate gradient operator on each component, thus penalizing only the undesired directional patterns.

B.3 Augmented Lagrangian method

The problem in (B.2) can be solved efficiently using the augmented Lagrangian approach [31, 42]. The idea consists of introducing intermediate variables $\mathbf{u}$ and transforming the unconstrained optimization problem in (B.2) into the equivalent constrained problem

$$\begin{aligned} &\underset{\mathbf{f},\mathbf{u}}{\text{minimize}} \ \frac{\mu}{2}\left\|\mathbf{H}\mathbf{f}-\mathbf{g}\right\|^2 + \left\|\mathbf{u}\right\|_{\text{ISO}} \\ &\text{subject to } \ \mathbf{u} = \mathbf{D}\mathbf{f}. \end{aligned} \tag{B.4}$$

The resulting problem is then solved using an augmented Lagrangian (AL) scheme [31, 42, 43]

$$L(\mathbf{f},\mathbf{u},\mathbf{y}) \ = \ \frac{\mu}{2}\left\|\mathbf{H}\mathbf{f}-\mathbf{g}\right\|^2 \ + \ \left\|\mathbf{u}\right\|_{\text{ISO}} \ - \ \mathbf{y}^{\text{T}}(\mathbf{u} \ - \ \mathbf{D}\mathbf{f}) \ + \ \frac{\rho}{2}\left\|\mathbf{u}-\mathbf{D}\mathbf{f}\right\|^2, \tag{B.5}$$

where ρ is a regularization parameter associated with the quadratic penalty term $\left\|\mathbf{u}-\mathbf{D}\mathbf{f}\right\|^2$, and $\mathbf{y}$ is the Lagrange multiplier associated with the constraint $\mathbf{u} = \mathbf{D}\mathbf{f}$.

The idea of the augmented Lagrangian method is to find a saddle point of $L(\mathbf{f},\mathbf{u},\mathbf{y})$ that is also the solution of the original problem (B.2). To this end, the alternating direction method of multipliers (ADMM) can be used to solve the following sub-problems iteratively [42]:

$$\hat{\mathbf{f}}_{q+1} = \arg\min_{\mathbf{f}} \ \frac{\mu}{2} \left\|\mathbf{Hf} - \mathbf{g}\right\|^2 - \hat{\mathbf{y}}_q^{\mathrm{T}}(\hat{\mathbf{u}}_q - \mathbf{Df}) + \frac{\rho}{2} \left\|\hat{\mathbf{u}}_q - \mathbf{Df}\right\|^2 \tag{B.6}$$

$$\hat{\mathbf{u}}_{q+1} = \arg\min_{\mathbf{u}} \ \left\|\mathbf{u}\right\|_{\mathrm{ISO}} - \hat{\mathbf{y}}_q^{\mathrm{T}}(\mathbf{u} - \mathbf{D}\hat{\mathbf{f}}_{q+1}) + \frac{\rho}{2} \left\|\mathbf{u} - \mathbf{D}\hat{\mathbf{f}}_{q+1}\right\|^2 \tag{B.7}$$

$$\hat{\mathbf{y}}_{q+1} = \hat{\mathbf{y}}_q - \rho(\hat{\mathbf{u}}_{q+1} - \mathbf{D}\hat{\mathbf{f}}_{q+1}). \tag{B.8}$$

Now, the **f**-subproblem in (B.6) has a closed-form solution and can be efficiently calculated through FFT [42]. The **u**-subproblem in (B.7) can be solved using the shrinkage formula [45] at very low cost, as well as the **y**-subproblem in (B.8), which consists of a mere update. For algorithmic details refer to Section B.7.

B.4 Problem Statement

We seek a method to improve the performance of traditional TV image deconvolution. If we know *a priori* that the image has some directional pattern, as in the case of Figure B.1, we can unbalance the amount of horizontal and vertical regularization towards the image content and thus we will get a better estimate. On the other hand, if the image does not exhibit a preferred pattern, e.g. Figure B.3, changing $\boldsymbol{\beta}$ does not provide any enhancement.

Still, the fact that different regions of Figure B.3 do possess directional patterns instigates us to use a similar method to enhance the results. Although Figures B.1 and B.3 are oversimplified and unrealistic, we will use them as didactic examples throughout this chapter. In fact, we can see them as exaggerated examples of the edges and patterns that occur in real images.

As a starting point for our developments, we will use the augmented Lagrangian method described in [42] for its state-of-the-art results, both in terms of image quality and speed.

B.4.1 Masking

A naïve attempt to solve this problem would be to mask different regions of the image and only then apply the appropriate regularization. This may be accomplished by introducing a masking operator **M** into the regularization term of (B.2) resulting in

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \ \frac{\mu}{2} \left\|\mathbf{Hf} - \mathbf{g}\right\|^2 + \left\|\mathbf{DMf}\right\|_{\mathrm{ISO}} \tag{B.9}$$

(a) $\mathbf{M}_x$: mask to be applied to Figure B.3 before using gradient operator $\mathbf{D}_x$.

(b) Result of the mask $\mathbf{M}_x$ applied to Figure B.3.

Figure B.6: Example of a tailor-made mask, which selects regions for horizontal gradient penalization.

with

$$\mathbf{DM} = \begin{bmatrix} \beta_x \mathbf{D}_x \mathbf{M}_x \\ \beta_y \mathbf{D}_y \mathbf{M}_y \end{bmatrix}. \tag{B.10}$$

Figure B.6a shows an example of mask which could be used when restoring an observed version of Figure B.3. Figure B.6b shows the masked image $\mathbf{M}_x\mathbf{f}$, on which only the horizontal gradient operator $\mathbf{D}_x$ should be applied.

Although this approach may yield good results, the effect on how (B.2) can be solved is quite negative. Specifically, the resulting matrix $\mathbf{DM}$ in (B.9) and (B.10) is no longer block circulant and thus fast FFT-based algorithms cannot be used. Due to large size of the variables involved, such consequence makes this approach impractical.

B.5 Pre-filtering/decomposing approach

Rather than using masks to select regions of the image that are better suited for a certain gradient operator, we will use pre-filters to perform such a task. The advantage of the proposed approach is that it maintains the block-circulant structure of the matrices involved allowing the use of fast algorithms.

Any image $\mathbf{f}$ can be decomposed as

$$\mathbf{f} = \mathbf{f}_x + \mathbf{f}_y, \tag{B.11}$$

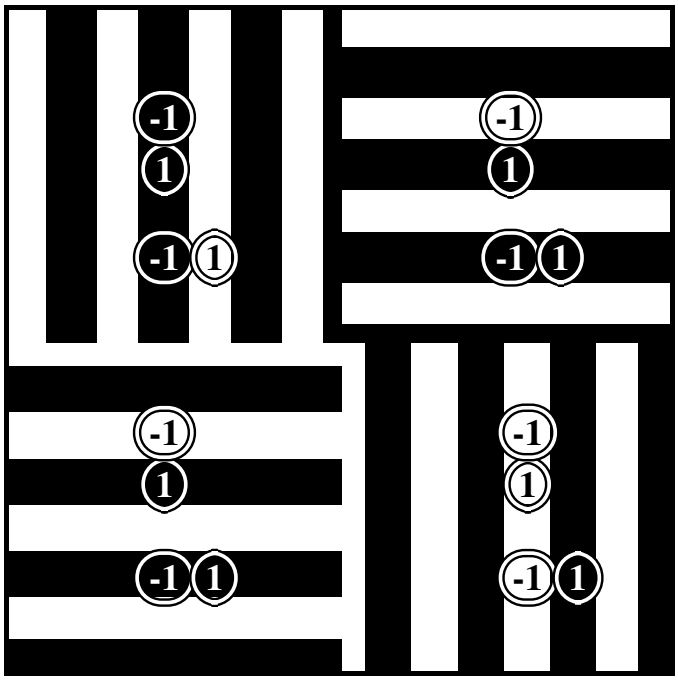


Figure B.7: An illustration of gradient calculation. Classical TV penalizes variations over the whole image in vertical (top) and horizontal (bottom circles) directions. For numbers with same colour, gradient is zero (null penalization), whereas different colours cause penalization. Thus, image patterns such as stripes are penalized.

where $\mathbf{f}_x$ represent the "horizontal" content or component of $\mathbf{f}$ and $\mathbf{f}_y$ "vertical" component. The components $\mathbf{f}_x$ and $\mathbf{f}_y$ are computed as

$$\mathbf{f}_x = \mathbf{B}_x\mathbf{f} \quad \text{and} \quad \mathbf{f}_y = \mathbf{B}_y\mathbf{f}, \tag{B.12}$$

where $\mathbf{B}_x$ and $\mathbf{B}_y$ are block-circulant matrices.

Combining (B.11) and (B.12) gives

$$\mathbf{f} = (\mathbf{B}_x + \mathbf{B}_y)\mathbf{f} \tag{B.13}$$

$$\mathbf{I} = \mathbf{B}_x + \mathbf{B}_y \tag{B.14}$$

and substituting (B.13) in (B.3) results in

$$\mathbf{D} = \mathbf{D}(\mathbf{B}_x + \mathbf{B}_y) \tag{B.15}$$

$$= \left(\begin{bmatrix} \beta_x \mathbf{D}_x \mathbf{B}_x \\ \beta_y \mathbf{D}_y \mathbf{B}_x \end{bmatrix} + \begin{bmatrix} \beta_x \mathbf{D}_x \mathbf{B}_y \\ \beta_y \mathbf{D}_y \mathbf{B}_y \end{bmatrix} \right). \tag{B.16}$$

We observe in (B.16) that the gradient operators are now applied to filtered versions of $\mathbf{f}$, though the effect has not changed due to identity $\mathbf{I} = \mathbf{B}_x + \mathbf{B}_y$. Since we want $\mathbf{D}_x$ to operate on the "horizontal" portion of $\mathbf{f}$ and $\mathbf{D}_y$ on its "vertical"

counterpart, we replace β_x and β_y which yields

$$\mathbf{D}_{2D} = \left(\begin{bmatrix} \frac{(1+\alpha)}{2}\mathbf{D}_x\mathbf{B}_x \\ \frac{(1-\alpha)}{2}\mathbf{D}_y\mathbf{B}_x \end{bmatrix} + \begin{bmatrix} \frac{(1-\alpha)}{2}\mathbf{D}_x\mathbf{B}_y \\ \frac{(1+\alpha)}{2}\mathbf{D}_y\mathbf{B}_y \end{bmatrix} \right) \tag{B.17}$$

with

$$0 \leq \alpha \leq 1, \tag{B.18}$$

where α controls the "adaptiveness". Now, when $\alpha = 0$ we have the traditional TV regularization equivalent to $\beta_x = \beta_y = 1$, whereas when $\alpha = 1$, $\mathbf{D}_x$ is applied only to $\mathbf{B}_x\mathbf{f}$ and $\mathbf{D}_y$ only to $\mathbf{B}_y\mathbf{f}$.

Simulations have shown that the choice of α should take into account the noise level. When noise is high, for instance, regularization should be less "adaptive" and setting α close to 1 will produce poor results. The intuition is that noise corrupts direction patterns, making it hard to select/filter for the use of a specific gradient in restoration.

Finally, the proposed algorithm is obtained by replacing $\mathbf{D}$ in equations (B.4) through (B.8) by $\mathbf{D}_{2D}$ in (B.17). Refer to Section B.7 for algorithmic details.

B.5.1 Choice of filters

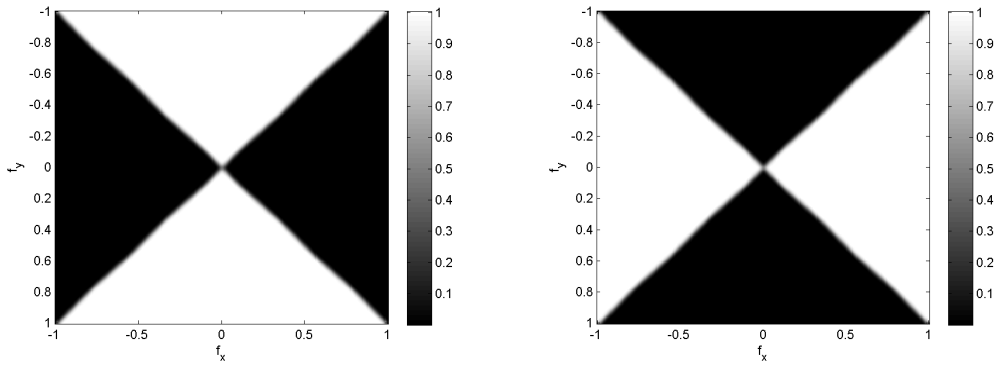
So far, we have introduced the idea of pre-filtering before applying regularization but have not defined the filters themselves. Following our initial assumptions, we know by (B.14) that the filters are complementary, i.e. their Fourier transforms add up to a constant.

According to Section B.1 and Figures B.1 and B.2, enhancement is possible when images have either horizontal or vertical orientation pattern. Therefore, we can design the filters by splitting the 2-D spectrum in horizontal ($|\omega_x| > |\omega_y|$) and vertical ($|\omega_x| < |\omega_y|$) frequencies and define the Fourier transforms of the filters from this partition as

$$\left|B_x(e^{j\omega_x}, e^{j\omega_y})\right| = \begin{cases} 0 & \text{if } |\omega_x| > |\omega_y| \\ \frac{1}{2} & \text{if } |\omega_x| = |\omega_y| \\ 1 & \text{if } |\omega_x| < |\omega_y| \end{cases} \tag{B.19}$$

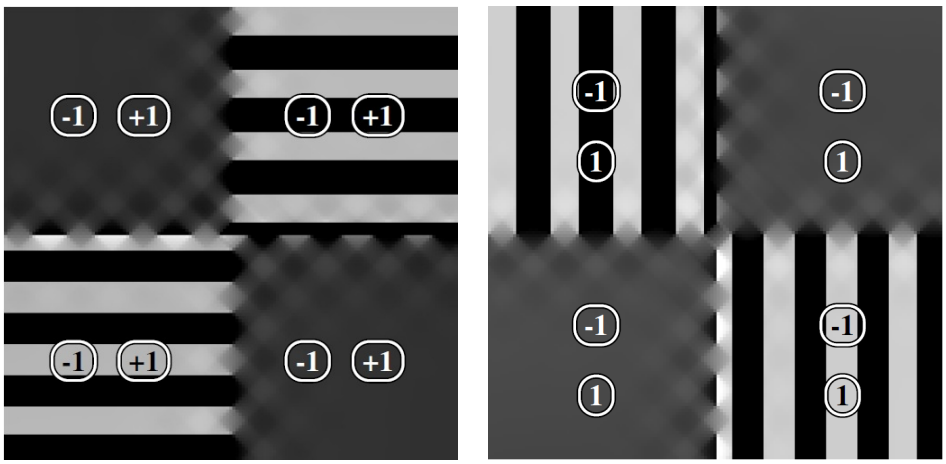
and $|B_y(e^{j\omega_x}, e^{j\omega_y})| = 1 - |B_x(e^{j\omega_x}, e^{j\omega_y})|$. Then, the filters can be designed through Matlab function `fwind1`, which uses the window method [111].

Figure B.9 illustrates the idea of the proposed method. After selecting regions of the image with filters, only the appropriate gradient is applied. Thus, image features are minimally penalized.



(a) $B_x(e^{j\omega_x}, e^{j\omega_y})$ (b) $B_y(e^{j\omega_x}, e^{j\omega_y})$

Figure B.8: Spectra of directional filters for the 2-direction deconvolution algorithm



(a) $\mathbf{B}_x\mathbf{f}$ (b) $\mathbf{B}_y\mathbf{f}$

Figure B.9: Gradients applied to image *components* in the proposed method. Penalization of image features is minimum since gradient is close to zero.

B.5.2 4-direction TV deconvolution

Simulations have shown that the approach we have just described is insufficient to enhance TV deconvolution for real images. We now extend the idea to incorporate diagonal gradients in addition to horizontal and vertical gradients by the definitions that follow.

The $\mathbf{D}_{4D}$ operator in the regularization term becomes

$$\mathbf{D}_{4D} = \left(\begin{bmatrix} \gamma\mathbf{D}_x\mathbf{B}_x \\ \delta\mathbf{D}_y\mathbf{B}_x \\ \delta\mathbf{D}_w\mathbf{B}_x \\ \delta\mathbf{D}_z\mathbf{B}_x \end{bmatrix} + \begin{bmatrix} \delta\mathbf{D}_x\mathbf{B}_y \\ \gamma\mathbf{D}_y\mathbf{B}_y \\ \delta\mathbf{D}_w\mathbf{B}_y \\ \delta\mathbf{D}_z\mathbf{B}_y \end{bmatrix} + \begin{bmatrix} \delta\mathbf{D}_x\mathbf{B}_w \\ \delta\mathbf{D}_y\mathbf{B}_w \\ \gamma\mathbf{D}_w\mathbf{B}_w \\ \delta\mathbf{D}_z\mathbf{B}_w \end{bmatrix} + \begin{bmatrix} \delta\mathbf{D}_x\mathbf{B}_z \\ \delta\mathbf{D}_y\mathbf{B}_z \\ \delta\mathbf{D}_w\mathbf{B}_z \\ \gamma\mathbf{D}_z\mathbf{B}_z \end{bmatrix} \right) \tag{B.20}$$

with

$$\gamma = 1 + \alpha, \quad \delta = 1 - \alpha \quad \text{and} \quad 0 \leq \alpha \leq 1. \tag{B.21}$$

The gradient matrices $\mathbf{D}_x$ through $\mathbf{D}_z$ perform the differences defined respectively by the kernels (filter coefficients)

$$d_x = \begin{bmatrix} -1 & 1 \end{bmatrix} \qquad d_y = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \tag{B.22}$$

$$d_w = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \qquad d_z = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{B.23}$$

The filters can be easily defined by partitioning the spectrum similarly to (B.19) and are summarized in Table B.1. As observed, the filters are more selective than those defined in Section B.5.1. Figure B.10 shows the spectra of the directional filters.

$\lvert B_x\rvert = 0$ if $\lvert 2\omega_x\rvert > \lvert\omega_y\rvert$, $\lvert B_x\rvert = 1$ otherwise
$\lvert B_y\rvert = 0$ if $\lvert\omega_x\rvert < \lvert 2\omega_y\rvert$, $\lvert B_y\rvert = 1$ otherwise
$\lvert B_w\rvert = 1$ if $\omega_x < 2\omega_y < 4\omega_x$ or $\omega_x > 2\omega_y > 4\omega_x$ $\lvert B_w\rvert = 0$ otherwise
$\lvert B_z\rvert = 1$ if $\omega_y < 2\omega_x < 4\omega_y$ or $\omega_y > 2\omega_x > 4\omega_y$ $\lvert B_z\rvert = 0$ otherwise

Table B.1: Filters for the 4-direction TV deconvolution. Additionally, $|B| = 1/2$ on the boundaries for all filters.

B.6 Results

In order to evaluate our method, we compared it to classical TV when deconvolving several blurred and noisy images. Figure B.11 shows our TV2D method applied to

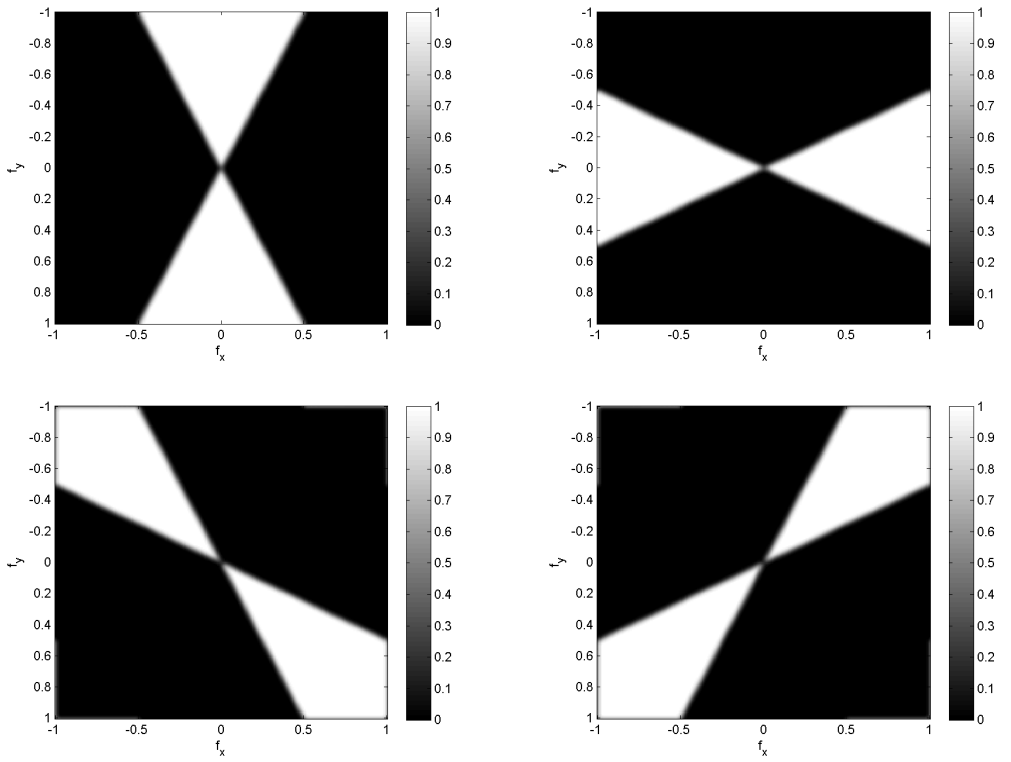


Figure B.10: Spectra of directional filters B_x, B_y, B_w and B_z used in the 4-direction algorithm

noisy and blurred versions of images on Figures B.1 and B.3 respectively. Figures B.12 and B.13 show the PSNR evolution of the proposed algorithm versus the classical approach.

Figures B.14 to B.17 show the PSNR evolution of the reconstructed image versus the regularization parameters μ for the classical TV and the proposed methods 2D and 4D. It is interesting to note that for Figure B.15 where vertical and horizontal orientations are pronounced, both 2-direction and 4-direction proposed algorithms provide similar results outperforming classical TV. For the other images which do not exhibit vertical/horizontal patterns, it is necessary to consider diagonal regularization in order to improve reconstruction, which is attained by the 4-direction method.

Figures B.18 through B.25 (zoomed-in versions of Cameraman, Mandrill and Lena, respectively) show real image results of the proposed 4-direction TV deconvolution algorithm compared to traditional TV deconvolution. Prior to restoration, the images were blurred with a 9×9 Gaussian kernel with $\sigma = 1.8$ and corrupted with noise $\mathcal{N}(0; 3 \times 10^{-5})$ (image dynamic range is $0 \sim 1$). The parameter α was empirically set to 0.5.

Table B.2 present some statistics of the penalization operators for the classical TV and the 4D proposed method. Calculations were performed over the 24 images of Kodak dataset. The proposed approach provides a better model for images, since

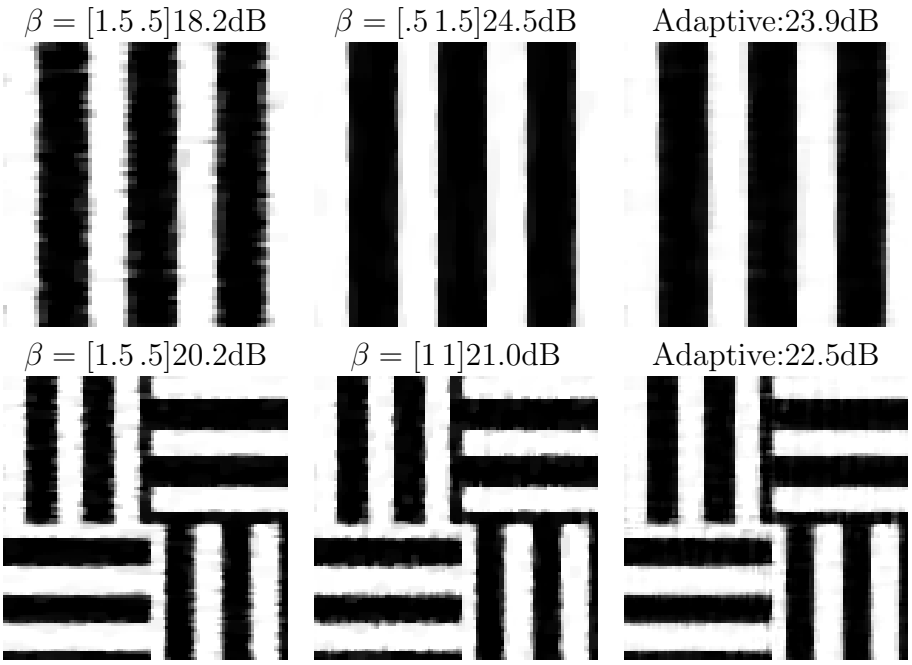


Figure B.11: Zoomed-in results of the proposed method (right column) compared with classical TV (left and center column). When the images are simple (top), choosing correct $\boldsymbol{\beta}$ is sufficient to improve reconstruction. For mixed patterns, changing $\boldsymbol{\beta}$ will improve only the appropriate regions (bottom left image).

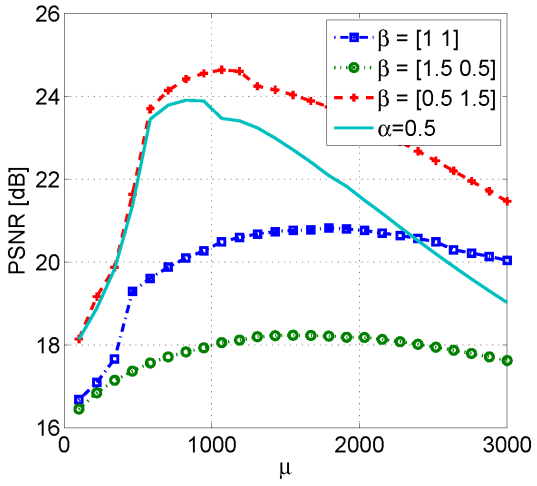


Figure B.12: Proposed method ($\alpha = 0.5$) provides results comparable to tuned classical TV restoration for the toy example of Figure B.1.

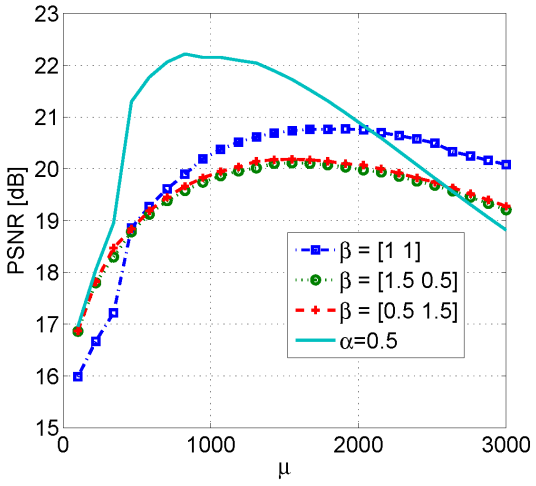


Figure B.13: For the mixed pattern of Figure B.3, the proposed method outperforms any combination of $\boldsymbol{\beta}$ in the classical TV.

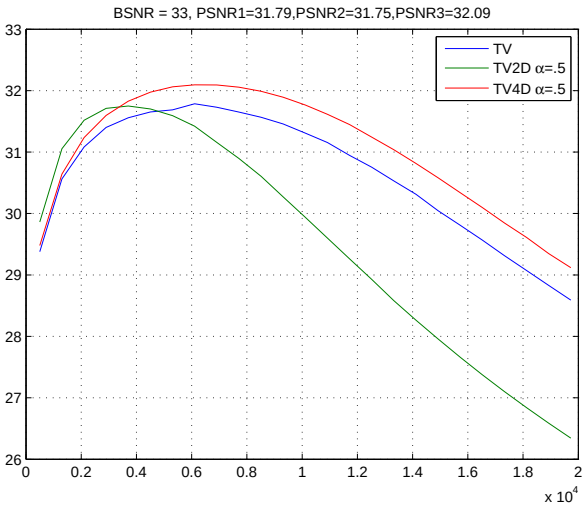


Figure B.14: Left: image used in the experiment - cameraman. Right: PSNR of the reconstructed image versus regularization parameter μ for Classical TV and proposed methods. TV2D uses two directional decomposition filters. TV4D uses four directional decomposition filters.

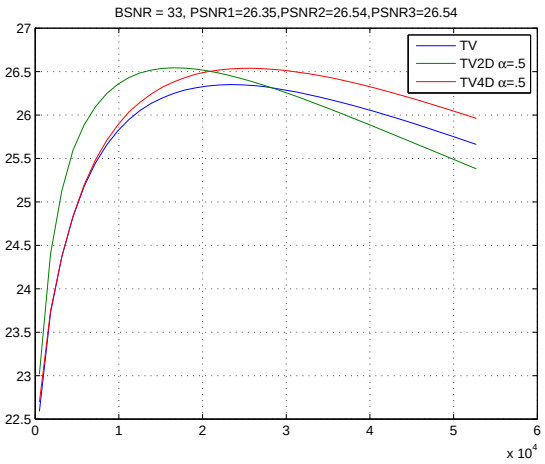


Figure B.15: Left: image used in the experiment - Kodak dataset 01. Right: PSNR of the reconstructed image versus regularization parameter μ for Classical TV and proposed methods. TV2D uses two directional decomposition filters. TV4D uses four directional decomposition filters.

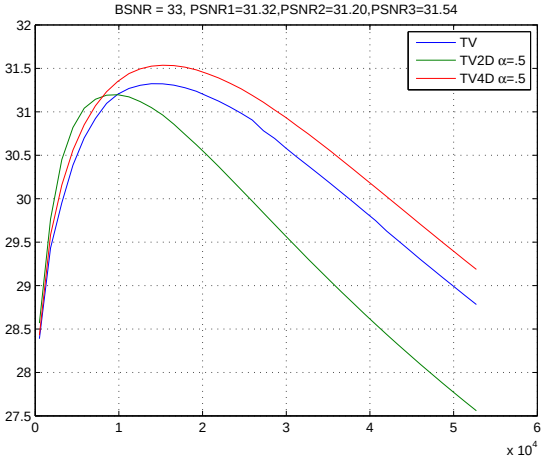


Figure B.16: Left: image used in the experiment - Kodak dataset 07. Right: PSNR of the reconstructed image versus regularization parameter μ for Classical TV and proposed methods. TV2D uses two directional decomposition filters. TV4D uses four directional decomposition filters.

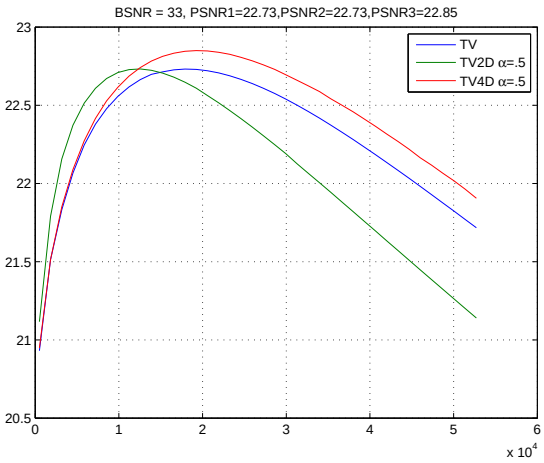


Figure B.17: Left: image used in the experiment - Kodak dataset 13. Right: PSNR of the reconstructed image versus regularization parameter μ for Classical TV and proposed methods. TV2D uses two directional decomposition filters. TV4D uses four directional decomposition filters.

Figure B.18: Classical TV best result: 83.15 dB

Figure B.19: Proposed 4D best result: 84.23 dB

Figure B.20: Classical TV best result: 80.86 dB

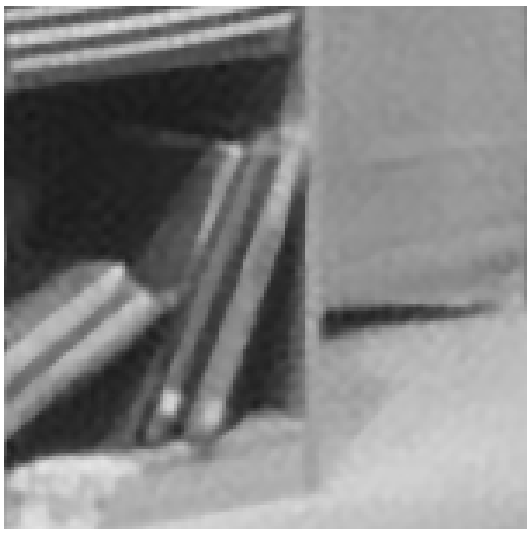


Figure B.21: Proposed 4D best result: 82.01 dB

Figure B.22: Classical TV best result: 76.92 dB

Figure B.23: Proposed 4D best result: 77.35 dB

Figure B.24: Classical TV best result: 77.74 dB

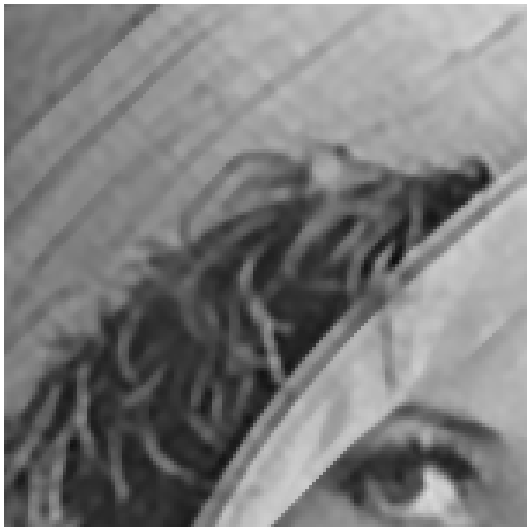


Figure B.25: Proposed 4D best result: 78.31 dB

Regularization	$\hat{\sigma}^2$ avg 10^3	$\hat{\sigma}^2$ var 10^6	$\hat{\phi}$ avg	$\hat{\phi}$ var 10^3
$\mathbf{D}_x\mathbf{f}$	3.7095	9.5098	0.4390	8.3591
$\mathbf{D}_y\mathbf{f}$	4.5040	9.5982	0.4552	11.1578
$\mathbf{D}_x\mathbf{B}_x\mathbf{f}$	0.1778	0.0214	0.5338	18.0484
$\mathbf{D}_y\mathbf{B}_y\mathbf{f}$	0.1105	0.0086	0.5627	23.9752
$\mathbf{D}_w\mathbf{B}_w\mathbf{f}$	0.1332	0.0102	0.5155	20.8042
$\mathbf{D}_z\mathbf{B}_z\mathbf{f}$	0.0955	0.0056	0.5832	19.8638

Table B.2: Statistics of traditional TV compared to directional filters + directional derivatives computed over the images from Kodak dataset.

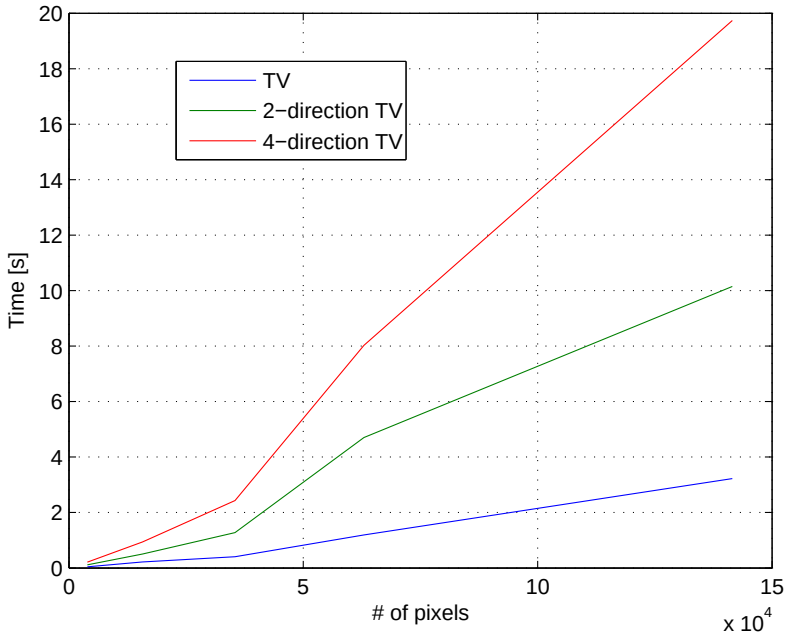


Figure B.26: Speed comparison of traditional TV and the directional decomposition TV approach

the average variance (model error) and variance of variance (consistency through different images) are lower. Moreover, the sample shape parameter $\hat{\phi}$ is closer to 1 in the proposed method, which indicates that the use of Laplacian prior is more appropriate than it is in classical TV.

Figure B.26 shows results of simulations performed to evaluate the speed of the algorithms. The same setup was used, but images were cropped to match the number of pixels desired. The results shown were obtained by averaging the time of 10 runs for each size under evaluation (images presented in the results were randomly selected to this experiment).

B.7 Algorithm details

First, we repeat here the pseudocode for the classical TV deconvolution described in [42].

Algorithm 1 Algorithm for TV/L2 minimization

Input data $\mathbf{g}$ and $\mathbf{H}$
Input parameters μ and $\boldsymbol{\beta}$
Set parameters ρ (default=2), θ (default=0.7) and ζ (default=2)
Initialize $\mathbf{f}_0 = \mathbf{g}$, $\mathbf{u}_0 = \mathbf{D}\mathbf{f}_0$, $\mathbf{y} = 0$, $k = 0$
Compute the matrices $\mathcal{F}[\mathbf{D}_x]$, $\mathcal{F}[\mathbf{D}_y]$ and $\mathcal{F}[\mathbf{H}]$
while not converge **do**
    Solve the $\mathbf{f}$-subproblem (B.24) using (B.27).
    Solve the $\mathbf{u}$-subproblem (B.25) using (B.28).
    Update the Lagrange multiplier $\mathbf{y}$ using (B.26).
    Update ρ according to (B.29).
    if $\|\mathbf{f}_{q+1} - \mathbf{f}_q\| / \|\mathbf{f}_q\| \leq \text{tol}$ **then**
        break ▷ Checking convergence
    end if
end while

The related sub-problems and the individual update equations are[2]

$$\hat{\mathbf{f}}_{q+1} = \arg\min_{\mathbf{f}} \ \frac{\mu}{2}\|\mathbf{H}\mathbf{f} - \mathbf{g}\|^2 - \hat{\mathbf{y}}_q^{\mathrm{T}}(\hat{\mathbf{u}}_q - \mathbf{D}\mathbf{f}) + \frac{\rho}{2}\|\hat{\mathbf{u}}_q - \mathbf{D}\mathbf{f}\|^2 \tag{B.24}$$

$$\hat{\mathbf{u}}_{q+1} = \arg\min_{\mathbf{u}} \ \|\mathbf{u}\|_{\mathrm{ISO}} - \hat{\mathbf{y}}_q^{\mathrm{T}}(\mathbf{u} - \mathbf{D}\hat{\mathbf{f}}_{q+1}) + \frac{\rho}{2}\left\|\mathbf{u} - \mathbf{D}\hat{\mathbf{f}}_{q+1}\right\|^2 \tag{B.25}$$

$$\hat{\mathbf{y}}_{q+1} = \hat{\mathbf{y}}_q - \rho(\hat{\mathbf{u}}_{q+1} - \mathbf{D}\hat{\mathbf{f}}_{q+1}). \tag{B.26}$$

$$\mathbf{f} = \mathcal{F}^{-1}\left[\frac{\mathcal{F}\left[\mu\mathbf{H}^{\mathrm{T}}\mathbf{g} + \rho\mathbf{D}^{\mathrm{T}}\mathbf{u} - \mathbf{D}^{\mathrm{T}}\mathbf{y}\right]}{\mu\left|\mathcal{F}[\mathbf{H}]\right|^2 + \rho\left(\left|\mathcal{F}[\mathbf{D}_x]\right|^2 + \left|\mathcal{F}[\mathbf{D}_y]\right|^2\right)}\right] \tag{B.27}$$

$$\begin{aligned}
\mathbf{v}_x &= \beta_x\mathbf{D}_x\mathbf{f} + (1/\rho)\mathbf{y}_x \\
\mathbf{v}_y &= \beta_y\mathbf{D}_y\mathbf{f} + (1/\rho)\mathbf{y}_y \\
\mathbf{v} &= \max\left\{\sqrt{|\mathbf{v}_x|^2 + |\mathbf{v}_y|^2}, \epsilon\right\} \quad \epsilon = 10^{-6} \\
\mathbf{u}_x &= \begin{cases} \max\{\mathbf{v} - 1/\rho, 0\} \cdot \left(\frac{\mathbf{v}_x}{\mathbf{v}}\right) & \text{isotropic TV} \\ \max\{|\mathbf{v}_x| - 1/\rho, 0\} \cdot \mathrm{sign}(\mathbf{v}_x) & \text{anisotropic TV} \end{cases}
\end{aligned} \tag{B.28}$$

[2]The Hadamard or element-wise division $\mathbf{p} = \left(\frac{\mathbf{q}}{\mathbf{r}}\right)$ will be used and is given by $[\mathbf{p}]_i = \frac{[\mathbf{q}]_i}{[\mathbf{r}]_i}$.

$$\rho = \begin{cases} \zeta\rho, & \text{if } \|\mathbf{u}_{q+1} - \mathbf{Df}_{q+1}\| \geq \theta \|\mathbf{u}_q - \mathbf{Df}_q\| \\ \rho, & \text{otherwise} \end{cases} \tag{B.29}$$

B.7.1 Proof of shrinkage formula

We provide here a proof of shrinkage formula (B.28) used to solve (B.25) for the case of anisotropic TV. For isotropic TV, similar approach can be used to derive the formula [45]. Proof of (B.27) is trivial and is omitted here. Updating ρ is suggested to accelerate convergence in [40] and is used in [42]. The proof for the convergence of the ADMM is shown in [44].

First we simplify (B.25) to

$$\mathbf{u}^* = \arg\min_{\mathbf{u}} \ \|\mathbf{u}\|_1 - \mathbf{y}^{\mathrm{T}}(\mathbf{u} - \mathbf{Df}) + \frac{\rho}{2}\|\mathbf{u} - \mathbf{Df}\|^2 . \tag{B.30}$$

We define the subdifferential of $\|\mathbf{u}\|_1$ given component-wise by [45, 112]

$$[\partial \|\mathbf{u}\|_1]_i = \begin{cases} \operatorname{sign}(u_i), & \text{if } u_i \neq 0 \\ \{h : |h| \leq 1, h \in \mathbf{R}\}, & \text{otherwise.} \end{cases} \tag{B.31}$$

From (B.30) and (B.31), each component u_i^* must satisfy

$$\begin{cases} \operatorname{sign}(u_i^*) - y_i + \rho(u_i^* - [\mathbf{Df}]_i) = 0, & \text{if } u_i^* \neq 0 \\ |y_i + \rho[\mathbf{Df}]_i| \geq 1, & \text{otherwise.} \end{cases} \tag{B.32}$$

For $u_i^* \neq 0$, we have

$$u_i^* + \frac{\operatorname{sign}(u_i^*)}{\rho} = [\mathbf{Df}]_i + \frac{y_i}{\rho} \tag{B.33}$$

and

$$|u_i^*| + \frac{1}{\rho} = \left| [\mathbf{Df}]_i + \frac{y_i}{\rho} \right| . \tag{B.34}$$

Setting

$$v_i = [\mathbf{Df}]_i + \frac{y_i}{\rho} \tag{B.35}$$

and dividing both equations we get

$$\frac{u_i^* + \operatorname{sign}(u_i^*)/\rho}{|u_i^*| + 1/\rho} = \operatorname{sign}(u_i^*) = \frac{v_i}{|v_i|} = \operatorname{sign}(v_i). \tag{B.36}$$

Therefore,

$$u_i^* = \left(|v_i| - \frac{1}{\rho} \right) \operatorname{sign}(v_i). \tag{B.37}$$

From (B.32) we know this is valid only when

$$|v_i| - \frac{1}{\rho} \geq 0, \tag{B.38}$$

otherwise we know that $u_i^* = 0$. Therefore

$$u_i^* = \max\left\{|v_i| - \frac{1}{\rho}, 0\right\} \operatorname{sign}(v_i). \tag{B.39}$$

Since all operations are done component-wise, we can group them in a vector-form

$$\mathbf{u}^* = \max\left\{|\mathbf{v}| - \frac{1}{\rho}, 0\right\} \operatorname{sign}(\mathbf{v}), \tag{B.40}$$

which is computationally very cheap.

B.8 Extension to directional filters

As previously mentioned, the proposed method replaces the classical operator $\mathbf{D} = [\mathbf{D}_x, \mathbf{D}_y]^{\mathrm{T}}$ in the previous equations by the directional operator

$$\mathbf{D}_{4D} = \left(\begin{bmatrix} \gamma\mathbf{D}_x\mathbf{B}_x \\ \delta\mathbf{D}_y\mathbf{B}_x \\ \delta\mathbf{D}_w\mathbf{B}_x \\ \delta\mathbf{D}_z\mathbf{B}_x \end{bmatrix} + \begin{bmatrix} \delta\mathbf{D}_x\mathbf{B}_y \\ \gamma\mathbf{D}_y\mathbf{B}_y \\ \delta\mathbf{D}_w\mathbf{B}_y \\ \delta\mathbf{D}_z\mathbf{B}_y \end{bmatrix} + \begin{bmatrix} \delta\mathbf{D}_x\mathbf{B}_w \\ \delta\mathbf{D}_y\mathbf{B}_w \\ \gamma\mathbf{D}_w\mathbf{B}_w \\ \delta\mathbf{D}_z\mathbf{B}_w \end{bmatrix} + \begin{bmatrix} \delta\mathbf{D}_x\mathbf{B}_z \\ \delta\mathbf{D}_y\mathbf{B}_z \\ \delta\mathbf{D}_w\mathbf{B}_z \\ \gamma\mathbf{D}_z\mathbf{B}_z \end{bmatrix} \right) \tag{B.41}$$

with

$$\gamma = 1 + \alpha, \quad \delta = 1 - \alpha, \quad 0 \leq \alpha \leq 1. \tag{B.42}$$

The gradient matrices $\mathbf{D}_x$ through $\mathbf{D}_z$ perform the differences defined respectively by the kernels (filter coefficients)

$$d_x = \begin{bmatrix} -1 & 1 \end{bmatrix} \qquad d_y = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \tag{B.43}$$

$$d_w = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \qquad d_z = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{B.44}$$

To save computer memory, we usually avoid the full construction of the matrices $\mathbf{D}$. Rather, we use the kernels to perform the convolutions in the frequency domain, which is advantageous in terms of speed and memory requirement.

Finally, what differs the proposed method from classical TV is how the direct

operators $\mathbf{D}_{4D}(\cdot)$ and $\mathbf{D}_{4D}^{\mathrm{T}}(\cdot)$ are calculated.

B.8.1 Direct operator

For classical TV, the operation $\mathbf{D}(\cdot)$ can be achieved by forward differences using, for instance, function `diff` in MATLAB.

In the proposed method, the operand has to be pre-filtered before the gradient operator application. Since both gradient operators and filters are fixed, we can combine them into new operators to save computational time. Thus, $\mathbf{D}_D\mathbf{f}$ is obtained by

$$\mathbf{D}_{4D}\mathbf{f} = \begin{bmatrix} \Delta_x\mathbf{f} \\ \Delta_y\mathbf{f} \\ \Delta_w\mathbf{f} \\ \Delta_z\mathbf{f} \end{bmatrix} \tag{B.45}$$

with

$$\begin{aligned} \Delta_x &= \gamma\mathbf{D}_x\mathbf{B}_x + \delta\mathbf{D}_x\mathbf{B}_y + \delta\mathbf{D}_x\mathbf{B}_w + \delta\mathbf{D}_x\mathbf{B}_z \\ \Delta_y &= \delta\mathbf{D}_y\mathbf{B}_x + \gamma\mathbf{D}_y\mathbf{B}_y + \delta\mathbf{D}_y\mathbf{B}_w + \delta\mathbf{D}_y\mathbf{B}_z \\ \Delta_w &= \delta\mathbf{D}_w\mathbf{B}_x + \delta\mathbf{D}_w\mathbf{B}_y + \gamma\mathbf{D}_w\mathbf{B}_w + \delta\mathbf{D}_w\mathbf{B}_z \\ \Delta_z &= \delta\mathbf{D}_z\mathbf{B}_x + \delta\mathbf{D}_z\mathbf{B}_y + \delta\mathbf{D}_z\mathbf{B}_w + \gamma\mathbf{D}_z\mathbf{B}_z. \end{aligned} \tag{B.46}$$

Since the convolutions are performed in the frequency domain, only the Fourier versions of the matrices Δ are stored.

B.8.2 Transposed operator

Since we need to use transpose operators, e.g. $\mathbf{D}_{4D}^{\mathrm{T}}(\cdot)$, but want to avoid dealing with full matrices, the effect of transposing a matrix on its defining kernel must be elucidated.

Simple matrix manipulation can show that transposing the kernels, e.g. $d_x \rightarrow d_x^{\mathrm{T}}$, is not enough to correctly build $\mathbf{D}_x^{\mathrm{T}}$. In fact, it can be easily shown that the operation $\mathbf{D}_x^{\mathrm{T}}\mathbf{u}$ can be obtained by a convolution using a 180-degree rotated version of d_x followed by a spatial shift proportional to the size of the kernel d_x, i.e.

$$\mathbf{D}_x^{\mathrm{T}}\mathbf{u} \equiv (u ** \mathrm{rot}_{180}(d_x))(x + n_c - 1, y + n_r - 1), \tag{B.47}$$

where n_r is the number of rows and n_c is the number of columns of d_x.

Therefore, we can define

$$\mathbf{D}_{4D}^{\mathrm{T}}\mathbf{u} = \begin{bmatrix} \Delta_x^{\mathrm{T}}\mathbf{u} \\ \Delta_y^{\mathrm{T}}\mathbf{u} \\ \Delta_w^{\mathrm{T}}\mathbf{u} \\ \Delta_z^{\mathrm{T}}\mathbf{u} \end{bmatrix} \tag{B.48}$$

where the transposition of the filter matrices $\mathbf{B}$ is obtained from the same recipe used in (B.47).

B.9 Conclusion

In this chapter, we proposed and showed results of an extension of the augmented Lagrangian approach [31, 42] for the problem of image deconvolution.

We started our development by noting that unbalancing the amount of horizontal and vertical regularization enhances classical TV deconvolution if the original image has a preferred frequency content, although the same is not true for more complex images.

In order to deal with a wider range of images, we recognized that different regions of the image require different regularizations. Rather than using masks to achieve this selection, we introduced directional decompositions/filters to perform this task. The advantage of the latter is that it permits the use of fast FFT-based algorithms due to block-circulant nature of matrices involved.

Even for images with a great deal of vertical or horizontal detail, our method has advantages because it does not require any previous knowledge about the image. For classical TV, on the other hand, one has to set the amount of horizontal and vertical regularization accordingly to the image.

In practical situations where the best regularization parameter μ is not available and has to be estimated, the proposed method presents another advantage of yielding better results for a wider range of μ. Specifically, even when the estimate of μ is not accurate, the proposed method has higher probability of generating better results than classical TV (see curves in Figures B.14 to B.17).

We presented experiments showing better PSNR and visual quality of the proposed method over the classical TV deconvolution.

We believe that increment in the running time of the proposed method can be reduced by avoiding "for" loops in our implementation in MATLAB. Interpreted languages are known to execute loops very slowly. One option is to write time-critical parts of the code in a compiled language like C.

Future work includes:

- Provide a deeper analysis of our method;

- Study computational complexity increment caused by increasing number of filters (e.g. more than 4 filters);
- Include a procedure for automatic parameter selection (parameters μ and α);
- Take advantage of the directional nature of our method and add an interpolation step/operation for extension to the problem of super-resolution/interpolation.
- Design the filters on-the-fly depending on the image content to privilege repetitive structures.
- Study the incorporation into our method of well-known directional decomposition, such as curvelets and counterlets.

■[3]

[3]Conduz-se o leitor de volta ao corpo da tese no Capítulo 4, página 26.

Appendix C

Preliminary studies on auto-regressive models

This chapter describes some preliminary studies and proposals for the use of piecewise autoregressive (PAR) models in inverse problems. The proposed methods consist of extensions of the *Soft-decision Adaptive Interpolation* (SAI) algorithm, which is described next.

C.1 Auto-regressive Modelling Approach

Zhang and Wu [59] proposed an image interpolation scheme that preserves spatial details. The technique adapts to varying scene structures using a 2-D piecewise autoregressive model. The model parameters are estimated in a moving window in the input low-resolution image. The algorithm is referred to as *Soft-decision Adaptive Interpolation* (SAI).

The SAI algorithm doubles the image resolution in both vertical and horizontal directions. It interpolates the missing pixels in HR (high resolution) image in two passes. Supposing an $N_1 \times N_2$ input image (Figure C.1), the first pass generates $(N_1 - 1) \times (N_2 - 1)$ pixels. These generated pixels along with the known LR (low resolution) pixels form a quincunx sublattice of the HR image (Figure C.2).

For the second pass, we first rotate the image obtained in the first pass by 45 degrees (Figure C.3). Then, we essentially reapply the first pass and complete the reconstruction of the HR image by interpolating the other quincunx sublattice (Figure C.4).

First, let's consider a PAR model of an image

$$g(\mathbf{n}) = \sum_{\mathbf{m} \in \mathcal{T}^H} \alpha_{\mathbf{m}}(\mathbf{n}) g(\mathbf{n} + \mathbf{m}) + \nu(\mathbf{n}), \quad \mathbf{n} = [n_1, n_2]^{\mathrm{T}}, \quad \mathbf{m} = [m_1, m_2]^{\mathrm{T}}, \qquad \text{(C.1)}$$

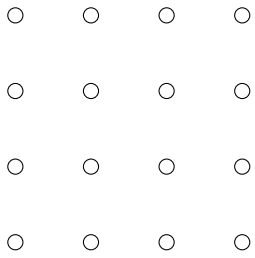


Figure C.1: Pixels of the LR original image in white.

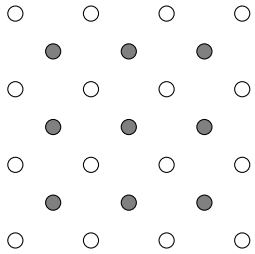


Figure C.2: Result of the first pass of SAI algorithm. Original pixels in white and estimated pixels in gray form a quincunx sublattice.

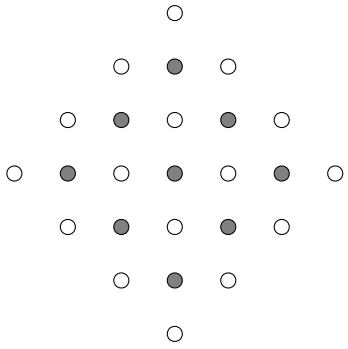


Figure C.3: Preparation for the second pass. If the image in Figure C.2 is rotated 45 degrees, we can reapply the first pass.

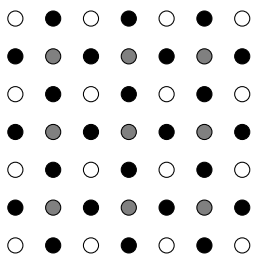


Figure C.4: Result of the second pass of SAI algorithm. Original pixels in white and gray and estimated pixels in black form a quincunx sublattice if the image is rotated 45 degrees.

$g(1)$		$g(4)$		$g(7)$
$g(2)$		$g(5)$		$g(8)$
$g(3)$		$g(6)$		$g(9)$

Figure C.5: Observed LR image pixels lexicographically ordered and placed on an HR grid

α_1		α_4		α_6
α_2		$g(\mathbf{n})$		α_7
α_3		α_5		α_8

Figure C.6: Auto-regressive model of order 8

where $\alpha_{\mathbf{m}}(\mathbf{n})$ are the model parameters, $\mathbf{n}$ and $\mathbf{m}$ are spatial coordinates, $\nu(\mathbf{n})$ is a random noise, and $\mathcal{T}^H$ is a spatial neighborhood template on the HR grid, which can be defined as

$$\begin{aligned}\mathcal{T}^H = \{ \quad & [-2,-2]^{\mathrm{T}}, \quad [-2,0]^{\mathrm{T}}, \quad [-2,2]^{\mathrm{T}}, \\ & [0,-2]^{\mathrm{T}}, \qquad\qquad [0,2]^{\mathrm{T}}, \\ & [2,-2]^{\mathrm{T}}, \qquad [2,0]^{\mathrm{T}} \qquad [2,2]^{\mathrm{T}} \quad \}.\end{aligned} \tag{C.2}$$

Figure C.5 shows the LR observed image placed on an HR grid. From now on, all the pictures will be shown in an HR grid unless otherwise specified.

Figure C.6 shows how a given pixel $g(\mathbf{n})$ can be modeled as a linear combination of its neighbors. The parameters $\alpha_{\mathbf{m}}(\mathbf{n})$ remain constant or nearly constant in a small neighborhood, though they may and often do vary significantly in different segments of a scene.

Instead of considering a model of order 8 as in (C.1), the SAI algorithm breaks it

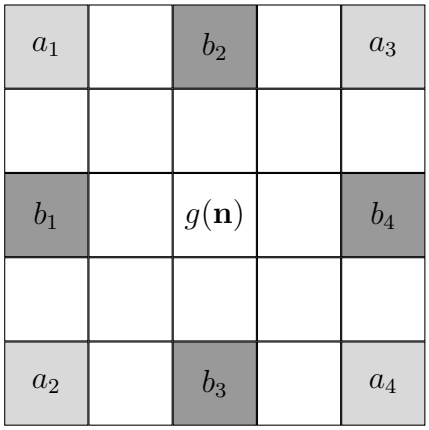


Figure C.7: Two auto-regressive models of order 4

in two parts. A given pixel $g(\mathbf{n})$ is estimated by two models, i.e.,

$$g(\mathbf{n}) = \sum_{\mathbf{m} \in \mathcal{T}_b^H} b_{\mathbf{m}} g(\mathbf{n} + \mathbf{m}) + \nu(\mathbf{n}), \tag{C.3}$$

and

$$g(\mathbf{n}) = \sum_{\mathbf{m} \in \mathcal{T}_a^H} a_{\mathbf{m}} g(\mathbf{n} + \mathbf{m}) + \nu(\mathbf{n}), \tag{C.4}$$

with

$$\begin{aligned} \mathcal{T}_b^H = \{ & \qquad\qquad [-2,0]^{\mathrm{T}}, \\ & [0,-2]^{\mathrm{T}}, \qquad\qquad [0,2]^{\mathrm{T}}, \\ & \qquad\qquad [2,0]^{\mathrm{T}} \qquad\qquad \}, \end{aligned} \tag{C.5}$$

and

$$\begin{aligned} \mathcal{T}_a^H = \{ \quad & [-2,-2]^{\mathrm{T}}, \qquad\qquad [-2,2]^{\mathrm{T}}, \\ & \\ & [2,-2]^{\mathrm{T}}, \qquad\qquad [2,2]^{\mathrm{T}} \quad \}. \end{aligned} \tag{C.6}$$

The two models are a diagonal cross (parameters a_i in Figure C.7 in lighter gray) and a vertical-horizontal cross (parameters b_i in Figure C.7 in darker gray). As an example, Figure C.8 shows the neighbors of pixel $g(5)$ according to each part of the model.

If we stack the parameters b_i and a_i we can form respectively two parameters vectors $\mathbf{b}$ and $\mathbf{a}$. Thus, these parameter vectors can be estimated in a training window W_1 of size typically 3×3 pixels by minimizing

(C.7)

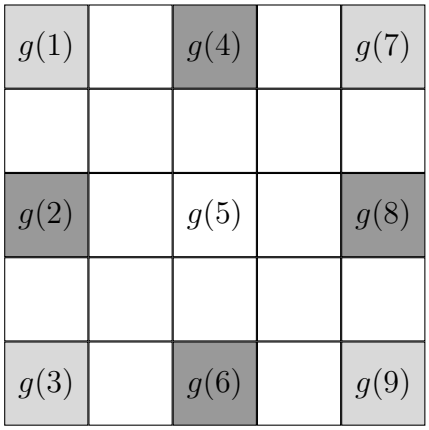


Figure C.8: Neighbors of pixel $g(5)$ according to the two models of SAI algorithm

$$\hat{\mathbf{b}} = \arg\min_{\mathbf{b}} \sum_{\mathbf{n}\in W_1} \left[g(\mathbf{n}) - \sum_{\mathbf{m}\in\mathcal{T}_b^H} b_{\mathbf{m}} g(\mathbf{n}+\mathbf{m}) \right]^2, \tag{C.8}$$

and

$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a}} \sum_{\mathbf{n}\in W_1} \left[g(\mathbf{n}) - \sum_{\mathbf{m}\in\mathcal{T}_a^H} a_{\mathbf{m}} g(\mathbf{n}+\mathbf{m}) \right]^2. \tag{C.9}$$

For instance, the window can be composed, on the HR grid, by (see Figure C.9)

$$\begin{aligned} W_1 = \{ \quad & [3,3]^{\mathrm{T}}, \quad [3,5]^{\mathrm{T}}, \quad [3,7]^{\mathrm{T}}, \\ & [5,3]^{\mathrm{T}}, \quad [5,5]^{\mathrm{T}} \quad [5,7]^{\mathrm{T}}, \\ & [7,3]^{\mathrm{T}}, \quad [7,5]^{\mathrm{T}} \quad [7,7]^{\mathrm{T}} \quad \}. \end{aligned} \tag{C.10}$$

Once the parameters $\mathbf{a}$ and $\mathbf{b}$ are obtained, we can proceed to the first pass of the SAI algorithm. The values for $f_1(\cdot)$ (number 1 stands for first pass) shown in Figure C.9, are obtained by minimizing

$$\begin{aligned} J(\lambda) = \min_{\mathbf{f}_1} \Bigg\{ & \sum_{\mathbf{n}\in W_2} \left\| f_1(\mathbf{n}) - \sum_{\mathbf{m}\in\mathcal{T}_a^L} a_{\mathbf{m}} g(\mathbf{n}+\mathbf{m}) \right\|^2 + \\ & \sum_{\mathbf{n}\in W_1} \left\| g(\mathbf{n}) - \sum_{\mathbf{m}\in\mathcal{T}_a^L} a_{\mathbf{m}} f_1(\mathbf{n}+\mathbf{m}) \right\|^2 + \\ & \lambda \sum_{\mathbf{n}\in W_3} \left\| f_1(\mathbf{n}) - \sum_{\mathbf{m}\in\mathcal{T}_b^H} b_{\mathbf{m}} f_1(\mathbf{n}+\mathbf{m}) \right\|^2 \Bigg\} \end{aligned} \tag{C.11}$$

	1	2	3	4	5	6	7	8	9
1	$g(1)$		$g(6)$		$g(11)$		$g(16)$		$g(21)$
2		$f_1(1)$		$f_1(5)$		$f_1(9)$		$f_1(13)$	
3	$g(2)$		$g(7)$		$g(12)$		$g(17)$		$g(22)$
4		$f_1(2)$		$f_1(6)$		$f_1(10)$		$f_1(14)$	
5	$g(3)$		$g(8)$		$g(13)$		$g(18)$		$g(23)$
6		$f_1(3)$		$f_1(7)$		$f_1(11)$		$f_1(15)$	
7	$g(4)$		$g(9)$		$g(14)$		$g(19)$		$g(24)$
8		$f_1(4)$		$f_1(8)$		$f_1(12)$		$f_1(16)$	
9	$g(5)$		$g(10)$		$g(15)$		$g(20)$		$g(25)$

Figure C.9: A typical window of the SAI algorithm. The LR pixels, represented by $g(\cdot)$, are lexicographically ordered. The HR pixels generated in the first pass $f_1(\cdot)$ are in lighter gray.

$$\text{subject to} \sum_{\mathbf{n}\in W_3} \left\| f_1(\mathbf{n}) - \sum_{\mathbf{m}\in\mathcal{T}_b^H} b_{\mathbf{m}} f_1(\mathbf{n}+\mathbf{m}) \right\|^2 \approx$$
$$\approx \sum_{\mathbf{n}\in W_3} \left\| g(\mathbf{n}) - \sum_{\mathbf{m}\in\mathcal{T}_b^H} b_{\mathbf{m}} g(\mathbf{n}+\mathbf{m}) \right\|^2 \tag{C.12}$$

where the parameter λ is chosen in (C.11) so that the approximation in (C.12) is satisfied.

The low-resolution template is given by

$$\begin{aligned} \mathcal{T}_a^L = \{ \quad & [-1,-1]^{\mathrm{T}}, && [-1,1]^{\mathrm{T}}, \\ & [1,-1]^{\mathrm{T}}, && [1,1]^{\mathrm{T}} \quad \}. \end{aligned} \tag{C.13}$$

The windows are given by

$$\begin{aligned} W_2 = \{ \quad & [2,2]^{\mathrm{T}}, \quad [2,4]^{\mathrm{T}}, \quad [2,6]^{\mathrm{T}}, \quad [2,8]^{\mathrm{T}}, \\ & [4,2]^{\mathrm{T}}, \quad [4,4]^{\mathrm{T}} \quad [4,6]^{\mathrm{T}}, \quad [4,8]^{\mathrm{T}}, \\ & [6,2]^{\mathrm{T}}, \quad [6,4]^{\mathrm{T}} \quad [6,6]^{\mathrm{T}}, \quad [6,8]^{\mathrm{T}}, \\ & [8,2]^{\mathrm{T}}, \quad [8,4]^{\mathrm{T}} \quad [8,6]^{\mathrm{T}}, \quad [8,8]^{\mathrm{T}}, \quad \}. \end{aligned} \tag{C.14}$$

$$\begin{aligned} W_3 = \{ \quad & [4,4]^{\mathrm{T}}, \quad [4,6]^{\mathrm{T}}, \\ & [6,4]^{\mathrm{T}}, \quad [6,6]^{\mathrm{T}} \quad \}. \end{aligned} \tag{C.15}$$

Experiments in [59] have shown that good results are obtained by simply choosing $\lambda = 0.5$. The cost function, therefore, becomes

$$\begin{aligned} \hat{\mathbf{f}}_1 = \arg\min_{\mathbf{f}_1} \Bigg\{ & \sum_{\mathbf{n}\in W_2} \left\| f_1(\mathbf{n}) - \sum_{\mathbf{m}\in\mathcal{T}_a^L} a_{\mathbf{m}} g(\mathbf{n}+\mathbf{m}) \right\|^2 + \\ & \sum_{\mathbf{n}\in W_1} \left\| g(\mathbf{n}) - \sum_{\mathbf{m}\in\mathcal{T}_a^L} a_{\mathbf{m}} f_1(\mathbf{n}+\mathbf{m}) \right\|^2 + \\ & 0.5 \sum_{\mathbf{n}\in W_3} \left\| f_1(\mathbf{n}) - \sum_{\mathbf{m}\in\mathcal{T}_b^H} b_{\mathbf{m}} f_1(\mathbf{n}+\mathbf{m}) \right\|^2 \Bigg\} \end{aligned} \tag{C.16}$$

Although all $f_1(\cdot)$ are obtained in Figure C.9 by minimizing (C.16), only the values for $f_1(6)$, $f_1(7)$, $f_1(10)$ and $f_1(11)$ are kept. Then, the window moves horizontally and vertically by 4 pixels for the next calculation and overlaps the previous step's window location. Zhang [59] argues that this overlapping procedure avoids undesired

artifacts in the final image.

By analyzing (C.16), we notice that the models in (C.8) and (C.9) and their parameters are assumed to be valid for the HR interpolated image, even though they are calculated from the LR image. For the parameter $\mathbf{b}$ in (C.8), this assumption is generally true for most natural images [59], since we use them to estimate pixels in the same scale, i.e., we use the values in the grid of $\mathbf{f}_1$ to estimate the pixels in the same $\mathbf{f}_1$.

On the other hand, the assumption that parameters $\mathbf{a}$ remain unchanged in different scales is more problematic, since the distance between the pixels used in (C.9) is twice the distance between the pixels used in (C.8). As argued in [58] this assumption holds if the window in question has edge(s) of a fixed orientation and of sufficiently large scale. However, experiments in [59] show that previous edge-based interpolation methods are prone to artifacts on small-scale spatial features of high curvature, for which the second order statistics may differ from LR to HR images. In such cases, the soft-decision estimation strategy of (C.11), i.e., minimizing estimation error for $f(\cdot)$ and $g(\cdot)$ from each other, can moderate those artifacts, making the proposed SAI approach considerably more robust and achieving unprecedented interpolation accuracy.

In [60], the property of a PAR model estimated in a LR grid be valid for the HR grid is referred to as *geometrical duality*, i.e., consistency of geometric structures across resolutions.

C.2 Proposed method: Modification in the Standard SAI Algorithm

In this section, we propose a modification in the standard SAI algorithm which enhances the results of the interpolation. We repeat the models and the objective function of SAI algorithm by convenience ($g(\mathbf{n})$ are the LR pixels and $f(\mathbf{n})$ are the HR pixels).

$$g(\mathbf{n}) = \sum_{\mathbf{m}\in\mathcal{T}_b^H} b_{\mathbf{m}} g(\mathbf{n}+\mathbf{m}) + \nu(\mathbf{n}), \tag{C.17}$$

$$g(\mathbf{n}) = \sum_{\mathbf{m}\in\mathcal{T}_a^H} a_{\mathbf{m}} g(\mathbf{n}+\mathbf{m}) + \nu(\mathbf{n}), \tag{C.18}$$

$$\hat{\mathbf{f}}_1 = \arg\min_{\mathbf{f}_1} \Bigg\{ \sum_{\mathbf{n}\in W_2} \Bigg\| f_1(\mathbf{n}) - \sum_{\mathbf{m}\in\mathcal{T}_a^L} a_{\mathbf{m}} g(\mathbf{n}+\mathbf{m}) \Bigg\|^2 + \sum_{\mathbf{n}\in W_1} \Bigg\| g(\mathbf{n}) - \sum_{\mathbf{m}\in\mathcal{T}_a^L} a_{\mathbf{m}} f_1(\mathbf{n}+\mathbf{m}) \Bigg\|^2 + 0.5 \sum_{\mathbf{n}\in W_3} \Bigg\| f_1(\mathbf{n}) - \sum_{\mathbf{m}\in\mathcal{T}_b^H} b_{\mathbf{m}} f_1(\mathbf{n}+\mathbf{m}) \Bigg\|^2 \Bigg\} \tag{C.19}$$

$$\mathcal{T}_b^H = \{ \quad [-2,0]^{\mathrm{T}}, \quad [0,-2]^{\mathrm{T}}, \quad [0,2]^{\mathrm{T}}, \quad [2,0]^{\mathrm{T}} \quad \}, \tag{C.20}$$

$$\mathcal{T}_a^H = \{ \quad [-2,-2]^{\mathrm{T}}, \quad [-2,2]^{\mathrm{T}}, \quad [2,-2]^{\mathrm{T}}, \quad [2,2]^{\mathrm{T}} \quad \}. \tag{C.21}$$

$$\mathcal{T}_a^L = \{ \quad [-1,-1]^{\mathrm{T}}, \quad [-1,1]^{\mathrm{T}}, \quad [1,-1]^{\mathrm{T}}, \quad [1,1]^{\mathrm{T}} \quad \}. \tag{C.22}$$

$$W_1 = \{ \quad [3,3]^{\mathrm{T}}, \quad [3,5]^{\mathrm{T}}, \quad [3,7]^{\mathrm{T}}, \quad [5,3]^{\mathrm{T}}, \quad [5,5]^{\mathrm{T}} \quad [5,7]^{\mathrm{T}}, \quad [7,3]^{\mathrm{T}}, \quad [7,5]^{\mathrm{T}} \quad [7,7]^{\mathrm{T}} \quad \}. \tag{C.23}$$

$$W_2 = \{ \quad [2,2]^{\mathrm{T}}, \quad [2,4]^{\mathrm{T}}, \quad [2,6]^{\mathrm{T}}, \quad [2,8]^{\mathrm{T}}, \quad [4,2]^{\mathrm{T}}, \quad [4,4]^{\mathrm{T}} \quad [4,6]^{\mathrm{T}}, \quad [4,8]^{\mathrm{T}}, \quad [6,2]^{\mathrm{T}}, \quad [6,4]^{\mathrm{T}} \quad [6,6]^{\mathrm{T}}, \quad [6,8]^{\mathrm{T}}, \quad [8,2]^{\mathrm{T}}, \quad [8,4]^{\mathrm{T}} \quad [8,6]^{\mathrm{T}}, \quad [8,8]^{\mathrm{T}}, \quad \}. \tag{C.24}$$

$$W_3 = \{ \quad [4,4]^{\mathrm{T}}, \quad [4,6]^{\mathrm{T}}, \quad [6,4]^{\mathrm{T}}, \quad [6,6]^{\mathrm{T}} \quad \}. \tag{C.25}$$

Conversely, we propose a modification that considers the whole image at once, rather than separated windows. Instead of using the same model parameters for all pixels within the window, we assign to each pixel in the image its own parameters, i.e.,

$$g(\mathbf{n}) = \sum_{\mathbf{m}\in\mathcal{T}_b^H} b_{\mathbf{m}}(\mathbf{n}) g(\mathbf{n}+\mathbf{m}) + \nu(\mathbf{n}), \tag{C.26}$$

and

$$g(\mathbf{n}) = \sum_{\mathbf{m}\in\mathcal{T}_a^H} a_{\mathbf{m}}(\mathbf{n}) g(\mathbf{n}+\mathbf{m}) + \nu(\mathbf{n}), \tag{C.27}$$

i.e., now $b_{\mathbf{m}}(\mathbf{n})$ and $a_{\mathbf{m}}(\mathbf{n})$ have the pixel index $\mathbf{n}$ (confront to (C.17) and (C.18)).

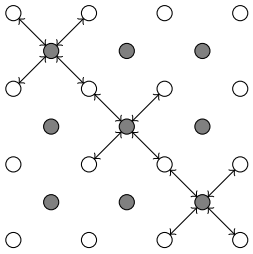


Figure C.10: Interdependence between top-left HR pixel and bottom-right HR pixel (in gray) through the various pixels of the image.

The justification is twofold: first, the auto-regressive parameters might slightly change within a window, and second, since we are obtaining the HR pixels minimizing the estimation error from their neighbors and to their neighbors (see (C.19)), the interdependence between LR and HR pixels is image-wide. In other words, the HR estimated pixel on the top-left of the image is related to the pixel located on the bottom-right of the image through the various auto-regressive models of the image.

The first step of the modification is to estimate the parameters $\mathbf{b}$ and $\mathbf{a}$ for all existing pixels in the LR image through equations (C.28) and (C.29), using a window of 3×3 pixels. That is, given a pixel $g(\mathbf{n})$, we obtain its related parameters through

$$\mathbf{b}(\mathbf{n}) = \arg\min_{\mathbf{b}} \sum_{\mathbf{q} \in W_4} \left[g(\mathbf{n}+\mathbf{q}) - \sum_{\mathbf{m} \in \mathcal{T}_b^H} b_{\mathbf{m}}(\mathbf{n}) g(\mathbf{n}+\mathbf{q}+\mathbf{m}) \right]^2, \tag{C.28}$$

and

$$\mathbf{a}(\mathbf{n}) = \arg\min_{\mathbf{a}} \sum_{\mathbf{q} \in W_4} \left[g(\mathbf{n}+\mathbf{q}) - \sum_{\mathbf{m} \in \mathcal{T}_a^H} a_{\mathbf{m}}(\mathbf{n}) g(\mathbf{n}+\mathbf{q}+\mathbf{m}) \right]^2. \tag{C.29}$$

with

$$\begin{aligned} W_4 = \{ \quad & [-2,-2]^{\mathrm{T}}, \quad [-2,0]^{\mathrm{T}}, \quad [-2,2]^{\mathrm{T}}, \\ & [0,-2]^{\mathrm{T}}, \quad [0,0]^{\mathrm{T}} \quad [0,2]^{\mathrm{T}}, \\ & [0,2]^{\mathrm{T}}, \quad [2,0]^{\mathrm{T}} \quad [2,2]^{\mathrm{T}} \quad \}. \end{aligned} \tag{C.30}$$

The vector $\mathbf{b}(\mathbf{n})$ is obtained lexicographically ordering the values $b_{\mathbf{q}}(\mathbf{n})$. Then, the window shifts vertically and horizontally by 1 pixel to calculate the parameters for the next pixel.

Once we have the parameters $\mathbf{b}$ and $\mathbf{a}$ for all existing pixels in the LR image, we obtain $\mathbf{b}$ and $\mathbf{a}$ for all HR pixels by simply averaging their LR neighbors. Figure C.11 shows an example of this calculation. The parameters obtained from the original pixels are referred to as $\mathbf{b}^g(\cdot)$ and the parameters for the HR pixels to be estimated in the first pass are referred to as $\mathbf{b}^f(\cdot)$. Figure C.12 shows the positions of original

$\mathbf{b}^g(1)$		$\mathbf{b}^g(5)$		$\mathbf{b}^g(9)$		$\mathbf{b}^g(13)$
	$\mathbf{b}^f(1)$		$\mathbf{b}^f(4)$		$\mathbf{b}^f(7)$	
$\mathbf{b}^g(2)$		$\mathbf{b}^g(6)$		$\mathbf{b}^g(10)$		$\mathbf{b}^g(14)$
	$\mathbf{b}^f(2)$		$\mathbf{b}^f(5)$		$\mathbf{b}^f(8)$	
$\mathbf{b}^g(3)$		$\mathbf{b}^g(7)$		$\mathbf{b}^g(11)$		$\mathbf{b}^g(15)$
	$\mathbf{b}^f(3)$		$\mathbf{b}^f(6)$		$\mathbf{b}^f(9)$	
$\mathbf{b}^g(4)$		$\mathbf{b}^g(8)$		$\mathbf{b}^g(12)$		$\mathbf{b}^g(16)$

Figure C.11: Estimation of model parameters for the unknown HR pixels of the first pass.

pixels $g(\cdot)$ and estimated pixels in the first pass $f(\cdot)$.

In this example, we have

$$\mathbf{b}^f(1) = \frac{\mathbf{b}^g(1) + \mathbf{b}^g(2) + \mathbf{b}^g(5) + \mathbf{b}^g(6)}{4}, \tag{C.31}$$

$$\mathbf{b}^f(2) = \frac{\mathbf{b}^g(2) + \mathbf{b}^g(3) + \mathbf{b}^g(6) + \mathbf{b}^g(7)}{4} \tag{C.32}$$

and so forth. The parameters $\mathbf{a}^f(\cdot)$ are also obtained by averaging their neighbors just as we do for $\mathbf{b}^f(\cdot)$.

We specify the parameters as $b_m^p(n)$ and $a_m^p(n)$, where $n = 1, 2, \cdots, N_1N_2$ denotes the pixel to which the parameter is related, $p = g$ denotes that the parameter was estimated from the original image and is related to it, $p = f$ denotes that the parameter was obtained by averaging and is related to the HR pixels and $m = 1, 2, 3, 4$ denotes the parameter number.

Suppose that the original image pixels are given by $g(\mathbf{n})$. If we lexicographically order the image, we get the vector representation $\mathbf{g}$. The pixels obtained in the first pass are also lexicographically arranged in the vector $\mathbf{f}$.

$g(1)$		$g(5)$		$g(9)$		$g(13)$
	$f(1)$		$f(4)$		$f(7)$	
$g(2)$		$g(6)$		$g(10)$		$g(14)$
	$f(2)$		$f(5)$		$f(8)$	
$g(3)$		$g(7)$		$g(11)$		$g(15)$
	$f(3)$		$f(6)$		$f(9)$	
$g(4)$		$g(8)$		$g(12)$		$g(16)$

Figure C.12: Original pixels $g(\cdot)$ and pixels to be estimated in the first pass of interpolation $f(\cdot)$.

We can use matrix notation to write (C.19) for the whole image as

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \left\{ \left\| \mathbf{f} - \mathbf{A}^f \mathbf{g} \right\|^2 + \left\| \mathbf{C}^g \mathbf{g} - \mathbf{A}^g \mathbf{f} \right\|^2 + 0.5 \left\| \mathbf{C}^f \mathbf{f} - \mathbf{B}^f \mathbf{f} \right\|^2 \right\} \tag{C.33}$$

If we take the example in Figures C.11 and C.12, the matrices become

$$\mathbf{A}^g = \begin{bmatrix} a_1^g(6) & a_2^g(6) & 0 & a_3^g(6) & a_4^g(6) & 0 & 0 & 0 & 0 \\ 0 & a_1^g(7) & a_2^g(7) & 0 & a_3^g(7) & a_4^g(7) & 0 & 0 & 0 \\ 0 & 0 & 0 & a_1^g(10) & a_2^g(10) & 0 & a_3^g(10) & a_4^g(10) & 0 \\ 0 & 0 & 0 & 0 & a_1^g(11) & a_2^g(11) & 0 & a_3^g(11) & a_4^g(11) \end{bmatrix} \tag{C.34}$$

$$(\mathbf{A}^f)^{\mathrm{T}} = \left[\begin{array}{ccccccccc}
a_1^f(1) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_2^f(1) & a_1^f(2) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & a_2^f(2) & a_1^f(3) & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & a_2^f(3) & 0 & 0 & 0 & 0 & 0 & 0 \\
a_3^f(1) & 0 & 0 & a_1^f(4) & 0 & 0 & 0 & 0 & 0 \\
a_4^f(1) & a_3^f(2) & 0 & a_2^f(4) & a_1^f(5) & 0 & 0 & 0 & 0 \\
0 & a_4^f(2) & a_3^f(3) & 0 & a_2^f(5) & a_1^f(6) & 0 & 0 & 0 \\
0 & 0 & a_4^f(3) & 0 & 0 & a_2^f(6) & 0 & 0 & 0 \\
0 & 0 & 0 & a_3^f(4) & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & a_4^f(4) & a_3^f(5) & 0 & a_1^f(7) & 0 & 0 \\
0 & 0 & 0 & 0 & a_4^f(5) & a_3^f(6) & a_2^f(7) & a_1^f(8) & 0 \\
0 & 0 & 0 & 0 & 0 & a_4^f(6) & 0 & a_2^f(8) & a_1^f(9) \\
0 & 0 & 0 & 0 & 0 & 0 & a_3^f(7) & 0 & a_2^f(9) \\
0 & 0 & 0 & 0 & 0 & 0 & a_4^f(7) & a_3^f(8) & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & a_4^f(8) & a_3^f(9) \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_4^f(9)
\end{array}\right] \tag{C.35}$$

$$(\mathbf{B}^g)^{\mathrm{T}} = \left[\begin{array}{cccc}
0 & 0 & 0 & 0 \\
b_1^g(6) & 0 & 0 & 0 \\
0 & b_1^g(7) & 0 & 0 \\
0 & 0 & 0 & 0 \\
b_2^g(6) & 0 & 0 & 0 \\
0 & b_2^g(7) & b_1^g(10) & 0 \\
b_3^g(6) & 0 & 0 & b_1^g(11) \\
0 & b_3^g(7) & 0 & 0 \\
0 & 0 & b_2^g(10) & 0 \\
b_4^g(6) & 0 & 0 & b_2^g(11) \\
0 & b_4^g(7) & b_3^g(10) & 0 \\
0 & 0 & 0 & b_3^g(11) \\
0 & 0 & 0 & 0 \\
0 & 0 & b_4^g(10) & 0 \\
0 & 0 & 0 & b_4^g(11) \\
0 & 0 & 0 & 0
\end{array}\right] \tag{C.36}$$

$$\mathbf{B}^f = \left[\begin{array}{cccccccc} 0 & b_1^f(5) & 0 & b_2^f(5) & 0 & b_3^f(5) & 0 & b_4^f(5) & 0 \end{array}\right] \tag{C.37}$$

$$\mathbf{C}^g = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{C.38}$$

$$\mathbf{C}^f = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{C.39}$$

For larger images, one has just to follow the same pattern to construct the matrices.

We can rearrange equation (C.33) by stacking the matrices and, then, we get

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \ \|\mathbf{K}\mathbf{f} - \mathbf{L}\mathbf{g}\|^2 \tag{C.40}$$

with

$$\mathbf{K} = \begin{bmatrix} \mathbf{I} \\ \mathbf{A}^g \\ 0.5(\mathbf{C}^f - \mathbf{B}^f) \end{bmatrix} \quad \text{and} \quad \mathbf{L} = \begin{bmatrix} \mathbf{A}^f \\ \mathbf{C}^g \\ \mathbf{0} \end{bmatrix} \tag{C.41}$$

which can be easily solved in a least-squares sense by

$$\hat{\mathbf{f}} = \left(\mathbf{K}^{\mathrm{T}}\mathbf{K}\right)^{-1}\mathbf{K}^{\mathrm{T}}\mathbf{L}\mathbf{g}. \tag{C.42}$$

Since we will later extend the Modified SAI algorithm to incorporate information from other frames, we simplify matrices $\mathbf{K}$ and $\mathbf{L}$ and the cost function to

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \ \|\mathbf{K}\mathbf{f} - \mathbf{L}\mathbf{g}\|^2 \tag{C.43}$$

with

$$\mathbf{K} = \begin{bmatrix} \mathbf{I} \\ \mathbf{A}^g \end{bmatrix} \quad \text{and} \quad \mathbf{L} = \begin{bmatrix} \mathbf{A}^f \\ \mathbf{C}^g \end{bmatrix}. \tag{C.44}$$

The reason for this simplification will become clear in Section C.4.1.

Due to the large size of the matrices $\mathbf{K}$ and $\mathbf{L}$, the direct least-squares solution in (C.42) is not efficient because of matrix inversion. Instead, since the matrices are also sparse, we adopt the biconjugate gradient stabilized method (BiCGstab(ℓ))[78, 113], thus avoiding matrix inversion. This method is widely used for solving large sparse unsymmetric linear systems and has provided good results in our experiments.

C.2.1 Results of the Modified SAI Algorithm

In order to assess the performance of the modified SAI algorithm, we blurred and decimated some frames from common videos and then applied four methods of

Image\Method	Cubic spline	GISD[114]	SAI[59]	Modified SAI
Carphone	29.6453	29.8620	29.8432	29.9672
Foreman	29.7510	29.9362	30.2352	30.0097
Miss america	38.2466	37.8825	37.9431	37.9738
Suzie	34.4888	34.4243	34.4902	34.5941

Table C.1: PSNR results of interpolation methods

Image\Method	Cubic spline	GISD[114]	SAI[59]	Modified SAI
Carphone	30.1728	30.5259	30.5350	30.6120
Foreman	30.6514	30.9595	31.0396	31.2311
Miss america	38.9363	38.5988	38.7313	38.5876
Suzie	35.0976	35.1027	35.2343	35.2690

Table C.2: PSNR results of interpolation methods after deblurring via BTV

interpolation: Cubic spline, GISD ([114]), SAI (see Section C.1) and the modified SAI. We used the following blurring kernel

$$\mathbf{h} = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}. \tag{C.45}$$

The PSNR's[1] between the original HR image and the estimated HR image for each method are shown in Table C.1. Note that no deblurring was performed for this experiment.

Table C.2 shows the results of interpolation with post deblurring via BTV method (a technique related to TV, see Appendix J). The parameters used were $It = 100$ (number of iterations), $\mu = 0.1$, $\lambda = 0.03$, $\alpha = 0.5$ and $P = 2$.

Figures C.13, C.14, C.15 and C.16 show the evolution of PSNR for the BTV deblurring of the selected frames.

C.3 Proposed method: Autoregressive Model Interpolation and Deblurring: Fusion and Deblurring in One Step

In this section, we propose a new algorithm which can perform interpolation and deblurring in one step and, thus, can achieve better results than other methods presented so far in this work.

[1] $\text{PSNR}(\mathbf{f}_1, \mathbf{g}_2) = 10 \log_{10} \frac{255^2}{\text{MSE}(\mathbf{f}_1, \mathbf{f}_2)}$

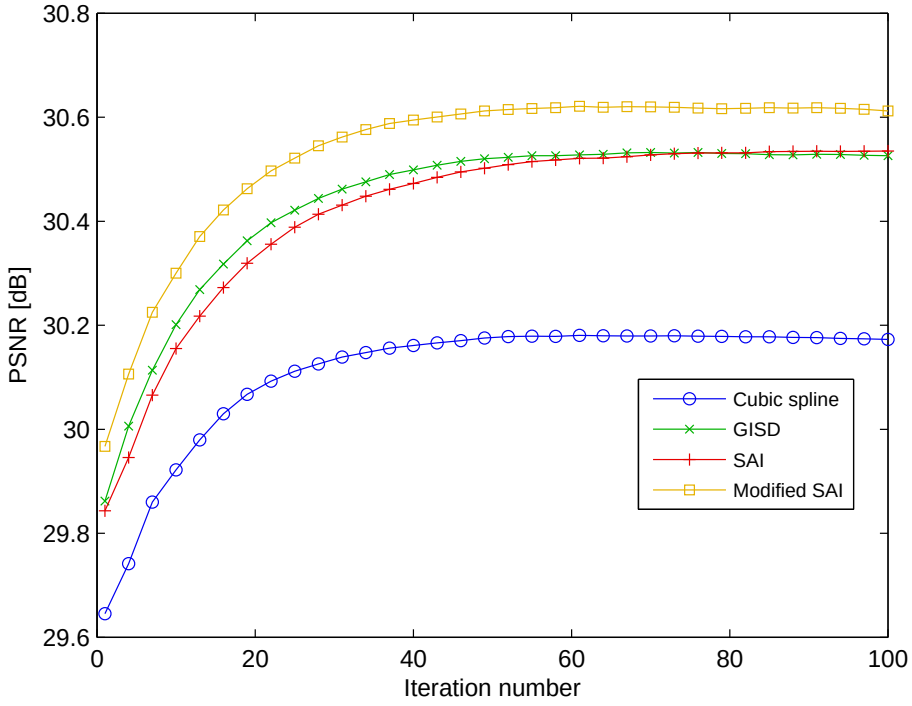


Figure C.13: Deblurring of Carphone video frame via BTV

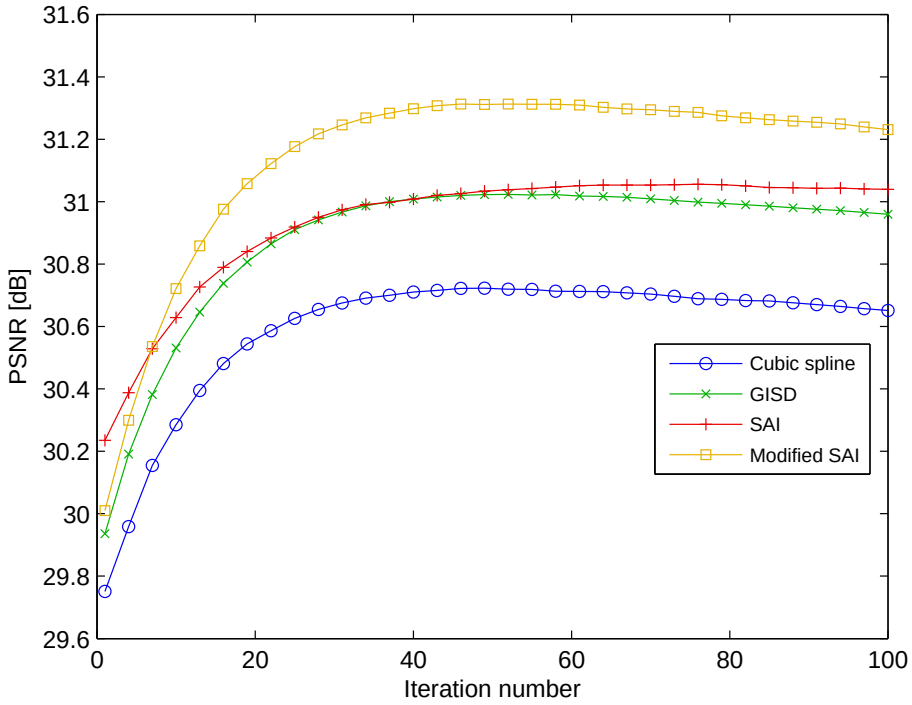


Figure C.14: Deblurring of Foreman video frame via BTV

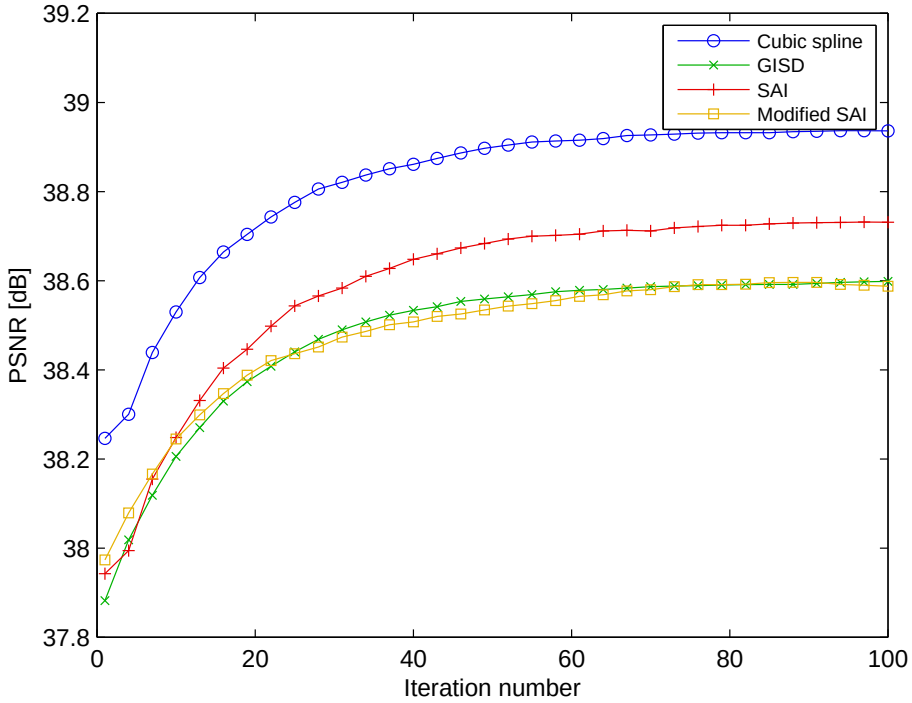


Figure C.15: Deblurring of Miss America video frame via BTV. Both SAI and Modified SAI are worse than cubic spline, probably because the large dark and almost flat area of the image. Imposing geometrical regularity to this area may have led to the failure.

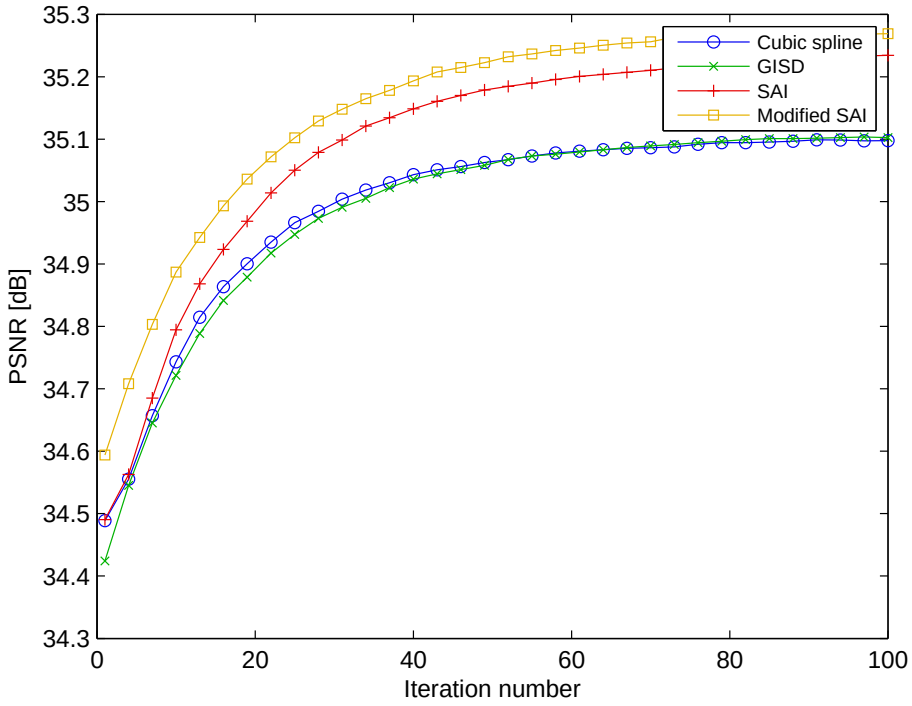


Figure C.16: Deblurring of Suzie video frame via BTV

C.3.1 Autoregressive Model Interpolation: AMI

First, we introduce an alternative method of interpolation, which is competitive with the traditional SAI algorithm. The proposed method is iterative and solves one sole cost function, rather than two cost functions as in the SAI and the Modified SAI algorithms. Then, we show how this method can be extended to perform deblurring. We will call this interpolation method *Autoregressive Model Interpolation* (AMI).

We return to the 8-element complete autoregressive model of a given image, say $g(\mathbf{n})$, repeated here by convenience

$$g(\mathbf{n}) = \sum_{\mathbf{m}\in\mathcal{T}^H} a_{\mathbf{m}}(\mathbf{n}) g(\mathbf{n}+\mathbf{m}) + \nu(\mathbf{n}), \quad \mathbf{n} = [n_1, n_2]^{\mathrm{T}}, \quad \mathbf{m} = [m_1, m_2]^{\mathrm{T}}, \tag{C.46}$$

where $a_{\mathbf{m}}(\mathbf{n})$ are the model parameters, $\mathbf{n}$ and $\mathbf{m}$ are spatial coordinates, $\nu(\mathbf{n})$ is a random noise, and $\mathcal{T}^H$ is a spatial neighborhood template on the HR grid, which can be defined as

$$\begin{aligned} \mathcal{T}^H = \{ \quad & [-2,-2]^{\mathrm{T}}, \quad [-2,0]^{\mathrm{T}}, \quad [-2,2]^{\mathrm{T}}, \\ & [0,-2]^{\mathrm{T}}, \qquad\qquad [0,2]^{\mathrm{T}}, \\ & [2,-2]^{\mathrm{T}}, \quad [2,0]^{\mathrm{T}} \quad [2,2]^{\mathrm{T}} \quad \}. \end{aligned} \tag{C.47}$$

As previously, we calculate the model parameters for all the pixels in the LR observed image by minimizing

$$\mathbf{a}(\mathbf{n}) = \arg\min_{\mathbf{a}(\mathbf{n})} \sum_{\mathbf{q}\in W_4} \left[g(\mathbf{n}+\mathbf{q}) - \sum_{\mathbf{m}\in\mathcal{T}^H} a_{\mathbf{m}}(\mathbf{n}) g(\mathbf{n}+\mathbf{q}+\mathbf{m}) \right]^2. \tag{C.48}$$

with

$$\begin{aligned} W_4 = \{ \quad & [-2,-2]^{\mathrm{T}}, \quad [-2,0]^{\mathrm{T}}, \quad [-2,2]^{\mathrm{T}}, \\ & [0,-2]^{\mathrm{T}}, \quad [0,0]^{\mathrm{T}} \quad [0,2]^{\mathrm{T}}, \\ & [0,2]^{\mathrm{T}}, \quad [2,0]^{\mathrm{T}} \quad [2,2]^{\mathrm{T}} \quad \}. \end{aligned} \tag{C.49}$$

Since we can only calculate the parameters for the LR pixels, we estimate the model parameters for the HR pixels by averaging their neighbors. For example, consider Figure C.17 where the parameters are placed on the HR grid. The LR pixels are in white, whereas the HR pixels are in gray. To obtain the HR parameter we do

$$\mathbf{a}(2) = \frac{\mathbf{a}(1) + \mathbf{a}(3)}{2} \tag{C.50}$$

$$\mathbf{a}(6) = \frac{\mathbf{a}(1) + \mathbf{a}(11)}{2} \tag{C.51}$$

$$\mathbf{a}(7) = \frac{\mathbf{a}(1) + \mathbf{a}(3) + \mathbf{a}(11) + \mathbf{a}(13)}{4} \tag{C.52}$$

$\mathbf{a}(1)$	$\mathbf{a}(6)$	$\mathbf{a}(11)$	$\mathbf{a}(16)$	$\mathbf{a}(21)$
$\mathbf{a}(2)$	$\mathbf{a}(7)$	$\mathbf{a}(12)$	$\mathbf{a}(17)$	$\mathbf{a}(22)$
$\mathbf{a}(3)$	$\mathbf{a}(8)$	$\mathbf{a}(13)$	$\mathbf{a}(18)$	$\mathbf{a}(23)$
$\mathbf{a}(4)$	$\mathbf{a}(9)$	$\mathbf{a}(14)$	$\mathbf{a}(19)$	$\mathbf{a}(24)$
$\mathbf{a}(5)$	$\mathbf{a}(10)$	$\mathbf{a}(15)$	$\mathbf{a}(20)$	$\mathbf{a}(25)$

Figure C.17: Estimating model parameters for HR pixels (gray) from the LR parameters (white)

and so forth. Thus, we obtain the model parameters for all pixels on the HR grid.

Now consider the following equation.

$$f(\mathbf{n}) = \sum_{\mathbf{m}\in\mathcal{T}^L} a_{\mathbf{m}}(\mathbf{n}) f(\mathbf{n}+\mathbf{m}), \tag{C.53}$$

with

$$\begin{aligned} \mathcal{T}^L = \{ \quad & [-1,-1]^{\mathrm{T}}, \quad [-1,0]^{\mathrm{T}}, \quad [-1,1]^{\mathrm{T}}, \\ & [0,-1]^{\mathrm{T}}, \qquad\qquad [0,1]^{\mathrm{T}}, \\ & [1,-1]^{\mathrm{T}}, \quad [1,0]^{\mathrm{T}} \quad [1,1]^{\mathrm{T}} \quad \}. \end{aligned} \tag{C.54}$$

Equation (C.53) calculates a pixel value $f(\mathbf{n})$ based on its neighbors. We can, thus, construct a matrix $\mathbf{A}$ that performs the calculations of equation (C.53) for all pixels in the pilot image by applying it to the vector $\mathbf{f}$, i.e.,

$$\mathbf{f}' = \mathbf{A}\mathbf{f}. \tag{C.55}$$

Suppose we want to measure how much the image $\mathbf{f}$ respects or is represented by the model $\mathbf{A}$. Therefore, we can calculate the following norm

$$\varepsilon_a = \|\mathbf{f} - \mathbf{A}\mathbf{f}\| \tag{C.56}$$

$$= \|(\mathbf{I} - \mathbf{A})\,\mathbf{f}\|\,. \tag{C.57}$$

We can now define a cost function as

$$\begin{aligned}\hat{\mathbf{f}} &= \arg\min_{\mathbf{f}} \ \frac{1}{2}\varepsilon_a^2 \\ &= \arg\min_{\mathbf{f}} \ \frac{1}{2}\left\|(\mathbf{I}-\mathbf{A})\,\mathbf{f}\right\|^2.\end{aligned} \tag{C.58}$$

As the SAI and the Modified SAI algorithms, we are assuming that the model $\mathbf{A}$, which is computed based on the LR pixels, remains valid for the HR pixels, mainly along edges. However, as shown by experiments, considering only the model $\mathbf{A}$ render the refinement of $\mathbf{f}$ prone to artifacts. Therefore, we add an extra term in the minimization.

Consider the equation

$$f(\mathbf{n}) = \sum_{\mathbf{m}\in\mathcal{T}^H} a_{\mathbf{m}}(\mathbf{n}) f(\mathbf{n}+\mathbf{m}), \tag{C.59}$$

with

$$\begin{aligned}\mathcal{T}^H = \{ \quad & [-2,-2]^{\mathrm{T}}, \quad [-2,0]^{\mathrm{T}}, \quad [-2,2]^{\mathrm{T}}, \\ & [0,-2]^{\mathrm{T}}, \qquad\qquad [0,2]^{\mathrm{T}}, \\ & [2,-2]^{\mathrm{T}}, \qquad [2,0]^{\mathrm{T}} \qquad [2,2]^{\mathrm{T}} \quad \}.\end{aligned} \tag{C.60}$$

Comparing to the equation (C.53), we changed the spatial template from its LR version $\mathcal{T}^L$ to its HR version $\mathcal{T}^H$.

Then, we construct a matrix $\mathbf{B}$ which performs the calculation of equation (C.59) as we did for matrix $\mathbf{A}$, i.e.,

$$\mathbf{f}'' = \mathbf{B}\mathbf{f}, \tag{C.61}$$

and the following norm measures how much the image $\mathbf{f}$ respects the model $\mathbf{B}$

$$\varepsilon_b = \|\mathbf{f}-\mathbf{B}\mathbf{f}\| \tag{C.62}$$

$$= \left\|(\mathbf{I}-\mathbf{B})\,\mathbf{f}\right\|. \tag{C.63}$$

We can now redefine the cost function as

$$\begin{aligned}\hat{\mathbf{f}} &= \arg\min_{\mathbf{f}} \ \frac{\lambda_a}{2}\varepsilon_a^2 + \frac{\lambda_b}{2}\varepsilon_b^2 \\ &= \arg\min_{\mathbf{f}} \ \frac{\lambda_a}{2}\left\|(\mathbf{I}-\mathbf{A})\,\mathbf{f}\right\|^2 + \frac{\lambda_b}{2}\left\|(\mathbf{I}-\mathbf{B})\,\mathbf{f}\right\|^2,\end{aligned} \tag{C.64}$$

where λ_a and λ_b are weighting factors.

We obtain an initial or pilot estimate for the interpolated image through a interpolation algorithm such as the SAI algorithm [59]

$$\hat{\mathbf{f}}_0 = \mathrm{SAI}\{\mathbf{g}\}. \tag{C.65}$$

Note that the pilot estimate is a lexicographically ordered image $\hat{\mathbf{f}}_0$. Then, we solve (C.64) by gradient descent, i.e.,

$$\hat{\mathbf{f}}_{q+1} = \hat{\mathbf{f}}_q - \mu \nabla_{\mathbf{f}} \varepsilon_q \tag{C.66}$$

with

$$\varepsilon_q = \frac{\lambda_a}{2} \left\| (\mathbf{I} - \mathbf{A}) \, \mathbf{f}_q \right\|^2 + \frac{\lambda_b}{2} \left\| (\mathbf{I} - \mathbf{B}) \, \mathbf{f}_q \right\|^2 \tag{C.67}$$

Applying the gradient to equation (C.67), we get the iterative formula

$$\begin{aligned} \mathbf{e}_a &= (\mathbf{I} - \mathbf{A}) \, \hat{\mathbf{f}}_q \\ \mathbf{e}_b &= (\mathbf{I} - \mathbf{B}) \, \hat{\mathbf{f}}_q \\ \hat{\mathbf{f}}_{q+1} &= \hat{\mathbf{f}}_q - \mu \mathbf{M} \left[\lambda_a \left(\mathbf{I} - \mathbf{A}^{\mathrm{T}} \right) \mathbf{e}_a + \lambda_b \left(\mathbf{I} - \mathbf{B}^{\mathrm{T}} \right) \mathbf{e}_b \right], \end{aligned} \tag{C.68}$$

where μ is the update step and the matrix $\mathbf{M}$ is a mask which restricts the updates to the HR pixels only. Since we are performing interpolation only, we do not want that the original LR pixels change theirs values.

The weighting factors λ_a and λ_b are chosen heuristically through the formulae

$$\lambda_a = \frac{1}{\varepsilon_a}, \qquad \lambda_b = \frac{1}{\varepsilon_b}. \tag{C.69}$$

Although we do not mathematically prove the choice of the weighting factors, we give an intuitive justification: each factor is inversely proportional to the estimation error, resulting in a degree of confidence in the model. If one model, say $\mathbf{A}$, has a small error, then $\mathbf{f}$ is well estimated and is in accordance with $\mathbf{A}$. Since we aim to maintain the characteristics of the image, we give more importance in (C.64) to the term which provides the smaller error. On the other hand, if the error ε_a is large, it means that the image cannot be well represented by $\mathbf{A}$. Therefore, the term will be given less importance in the cost function.

Results of AMI Algorithm

Since we are evaluating the performance of algorithms in terms of interpolation only, we generate LR images $g(\mathbf{n})$ from the original image $f(\mathbf{n})$ by simply decimating it, i.e.,

$$g(\mathbf{n}) = f(2\mathbf{n}) \quad \mathbf{n} = [n_1, n_2]^{\mathrm{T}}. \tag{C.70}$$

Initial results of the AMI algorithm showed that it suffers from overfitting. Figures C.18, C.19, C.20 and C.21 show that the PSNR starts to decrease after a certain number of iterations. Those figures also show the evolution of the Total Variation (TV) of the image with the number of steps.

Based on experiments, we propose a stop criteria based on the TV of the image.

Method\Video	Carphone	Foreman	Miss America	Suzie
SAI	30.5299	33.3421	39.0789	35.3770
Modified SAI	29.3178	32.0757	37.7048	34.3597
AMI max	30.5466	33.3652	39.0900	35.4196
AMI TV stop	30.5465	33.3651	39.0856	35.4192

Table C.3: PSNR results of AMI algorithm (interpolation only)

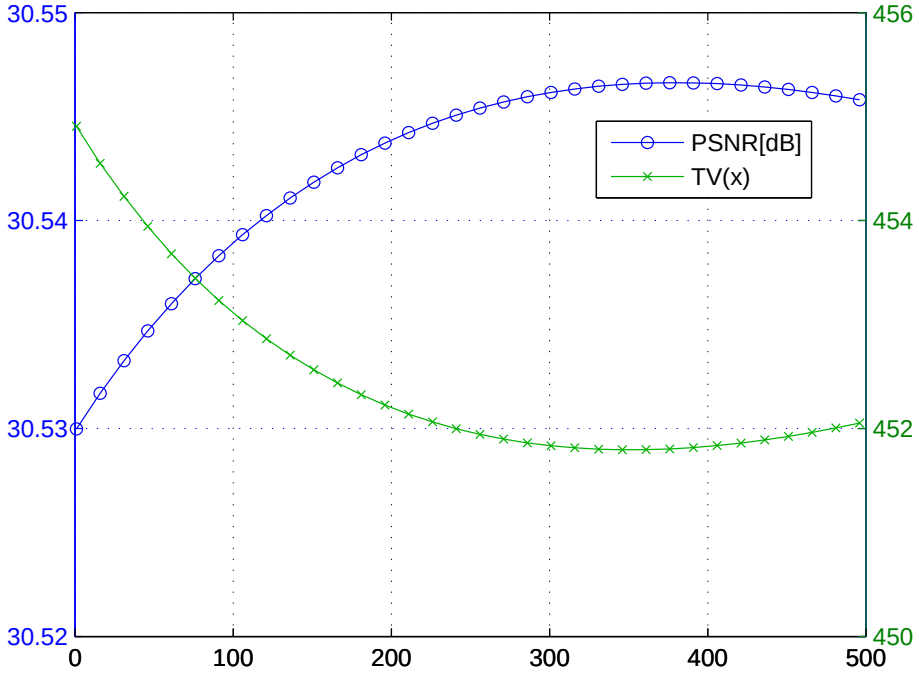


Figure C.18: PSNR of AMI algorithm evolution for Carphone video frame along with TV stop criteria

Unlike the PSNR which needs the real image as reference, TV can be measured directly from the $\hat{\mathbf{f}}_q$. Thus, when the TV changes its derivative from negative to positive, we stop the algorithm.

Table C.3 shows comparative results of four interpolation methods. We show the best result for the AMI algorithm and the PSNR obtained with the TV stop criteria.

C.3.2 Adding Deblurring Term: AMID

So far, we have presented the AMI algorithm as a refinement for interpolation methods. However, we did not consider blurring before decimation of the images in the formulation of the problem. We propose now a new algorithm for interpolation and deblurring in one step, which will be called *Autoregressive Model Interpolation and Deblurring* (AMID).

Since most images and video sequences are blurred before decimation, we add an

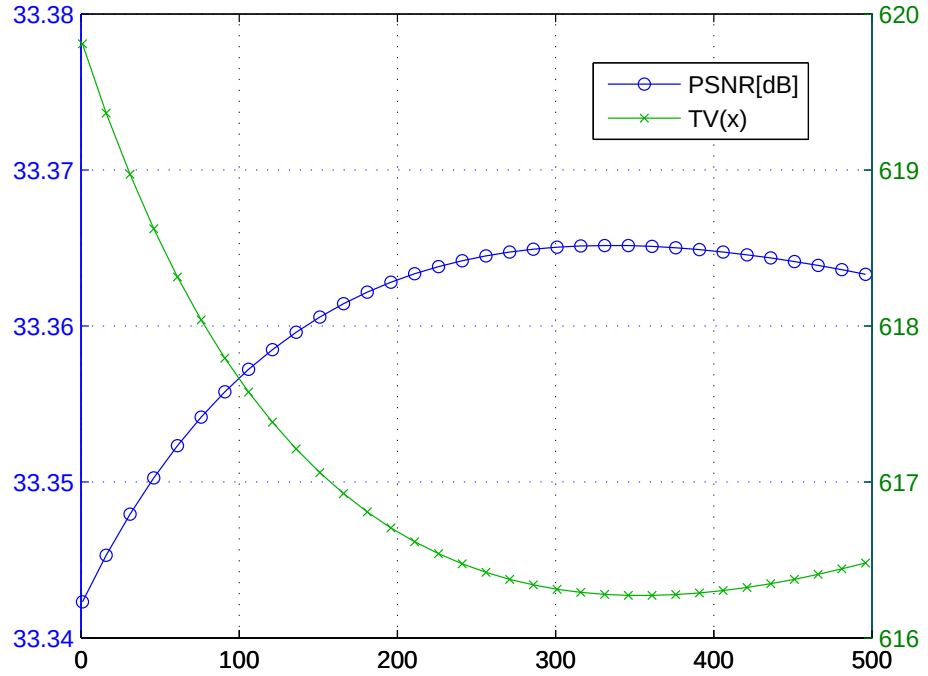


Figure C.19: PSNR of AMI algorithm evolution for Foreman video frame along with TV stop criteria

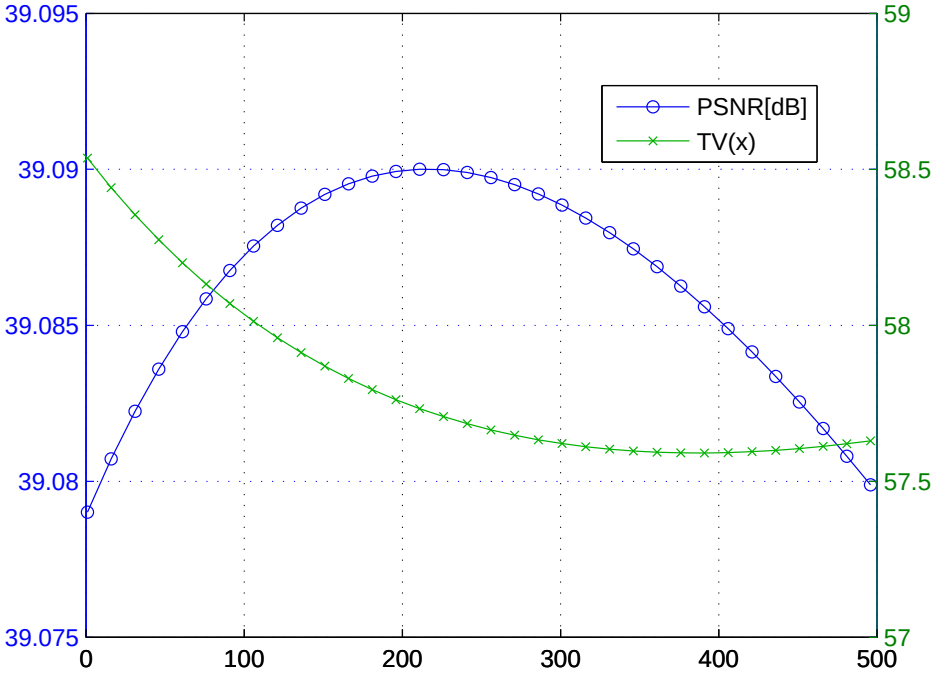


Figure C.20: PSNR of AMI algorithm evolution for Miss America video frame along with TV stop criteria

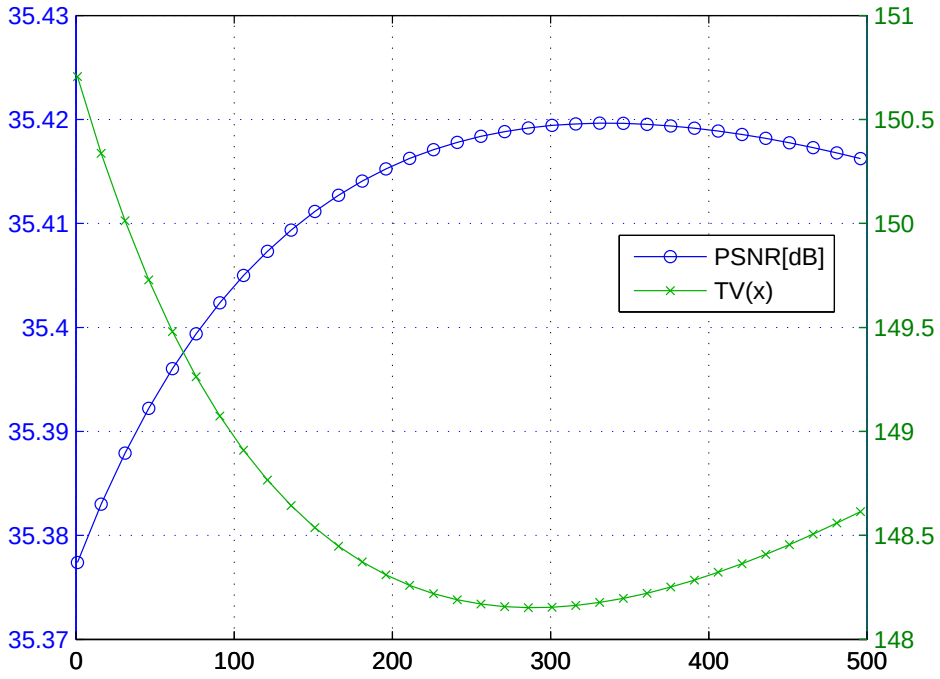


Figure C.21: PSNR of AMI algorithm evolution for Suzie video frame along with TV stop criteria

extra term to the equation (C.64) resulting in

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \; \|\mathbf{RHf} - \mathbf{g}\|^2 + \frac{\lambda}{\lambda_a + \lambda_b}\Big[\lambda_a \|(\mathbf{I} - \mathbf{A})\,\mathbf{f}\|^2 + \lambda_b \|(\mathbf{I} - \mathbf{B})\,\mathbf{f}\|^2\Big]. \quad \text{(C.71)}$$

The extra term, $\|\mathbf{RHf} - \mathbf{g}\|^2$ accounts for the conformity or agreement between the observed image $\mathbf{g}$ and the estimate $\mathbf{f}$ through the observation operators $\mathbf{RH}$. The multiplier λ balances the amount of regularization desired.

What we are actually proposing in (C.71) is a deblurring method based on a regularization term which is calculated according to the image content. Unlike Total Variation or Bilateral Total Variation which aim to minimize the total energy of image gradients, our regularization term aims to minimize the disaccordance between the estimated image and the autoregressive models $\mathbf{A}$ and $\mathbf{B}$.

Method\Video	Carphone	Foreman	Miss America	Suzie
SAI & BTV	31.6304	33.2643	39.1750	35.9866
Modified SAI & BTV	30.9434	31.7586	37.9344	34.9883
BTV reconstruction	31.6278	33.1783	39.7281	36.0959
AMID	31.8205	33.9584	40.0099	36.3040
AMID gain over SAI+BTV	0.1901	0.6941	0.8349	0.3174

Table C.4: PSNR results of AMID algorithm (interpolation and deblurring)

Solving equation (C.71), we have the iterative formula

$$\begin{aligned}
\mathbf{e}_h &= \mathbf{R}\mathbf{H}\hat{\mathbf{f}}_q - \mathbf{g} \\
\mathbf{e}_a &= (\mathbf{I} - \mathbf{A})\,\hat{\mathbf{f}}_q \\
\mathbf{e}_b &= (\mathbf{I} - \mathbf{B})\,\hat{\mathbf{f}}_q \\
\hat{\mathbf{f}}_{q+1} &= \hat{\mathbf{f}}_q - \mu\Big\{\mathbf{H}^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}\mathbf{e}_h + \\
&\qquad + \frac{\lambda}{\lambda_a + \lambda_b}\left[\lambda_a\left(\mathbf{I} - \mathbf{A}^{\mathrm{T}}\right)\mathbf{e}_a + \lambda_b\left(\mathbf{I} - \mathbf{B}^{\mathrm{T}}\right)\mathbf{e}_b\right]\Big\}.
\end{aligned} \tag{C.72}$$

Results of AMID Algorithm

In order to assess the performance of the AMID algorithm, we blurred and decimated several frames of well-known videos and applied several methods of interpolation and deblurring.

We empirically adjusted the parameters of BTV deblurring algorithm to achieve the best performance in terms of PSNR. The new parameters are $It = 150$ (number of iterations), $\mu = 0.1$, $\lambda = 1e-5$, $\alpha = 0.5$ and $P = 2$. For the AMID algorithm the parameters used were: $\lambda = 5e-3$ and $\mu = 2$.

We also compared the performance of AMID algorithm to the BTV reconstruction algorithm given by

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \left\{ \|\mathbf{R}\mathbf{H}\mathbf{f} - \mathbf{g}\| + \lambda \underbrace{\sum_{l=-P}^{P}\sum_{m=0}^{P}}_{l+m\geq 0} \alpha^{|m|+|l|} \left\|\mathbf{f} - \mathbf{S}_x^l\mathbf{S}_y^m\mathbf{f}\right\|_1 \right\}. \tag{C.73}$$

Table C.4 shows the PSNR results for some deblurring strategies. Figures C.22, C.23, C.24 and C.25 shows the PSNR evolution with the iteration number. Note that we also show the TV of the image. Although the TV increases at first iterations, it tends to stabilize or increase less rapidly with the iteration number.

Figures C.26, C.27, C.28 and C.29 show comparative results for SAI+BTV deblurring and our AMID algorithm.

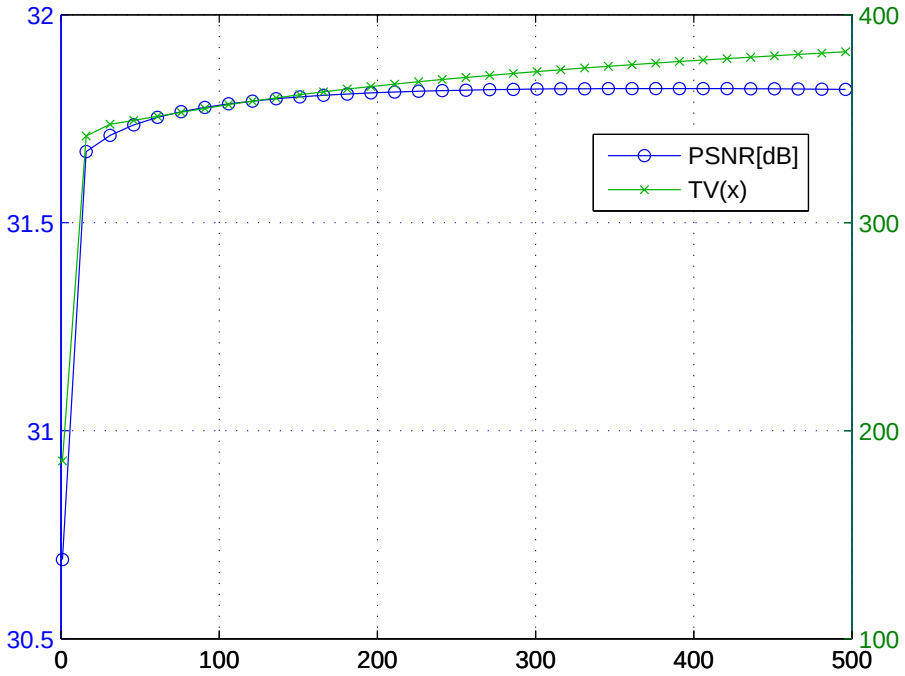


Figure C.22: PSNR of AMID algorithm evolution for Carphone video frame with TV

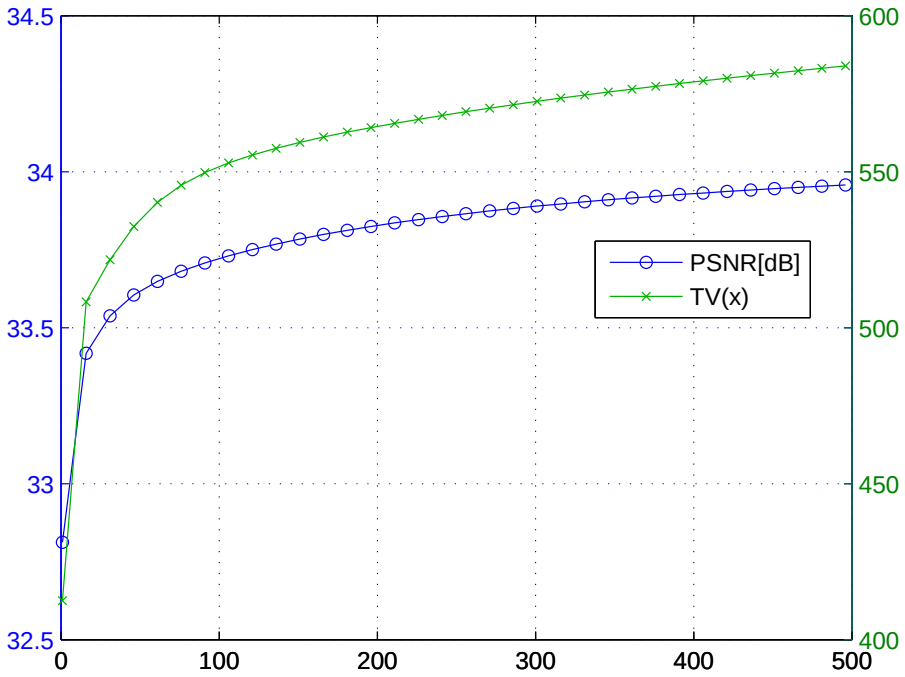


Figure C.23: PSNR of AMID algorithm evolution for Foreman video frame along with TV

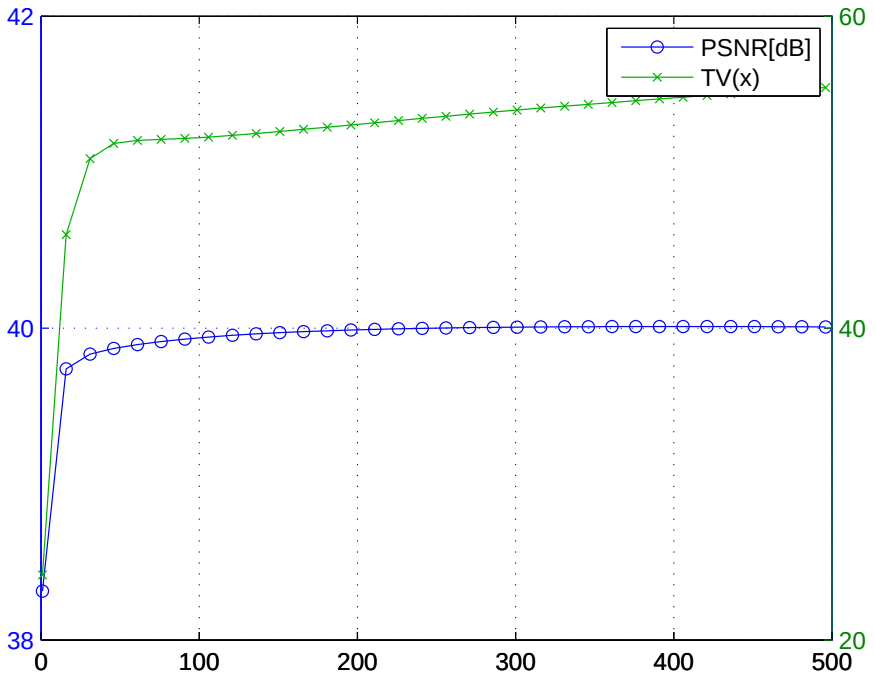


Figure C.24: PSNR of AMID algorithm evolution for Miss America video frame along with TV

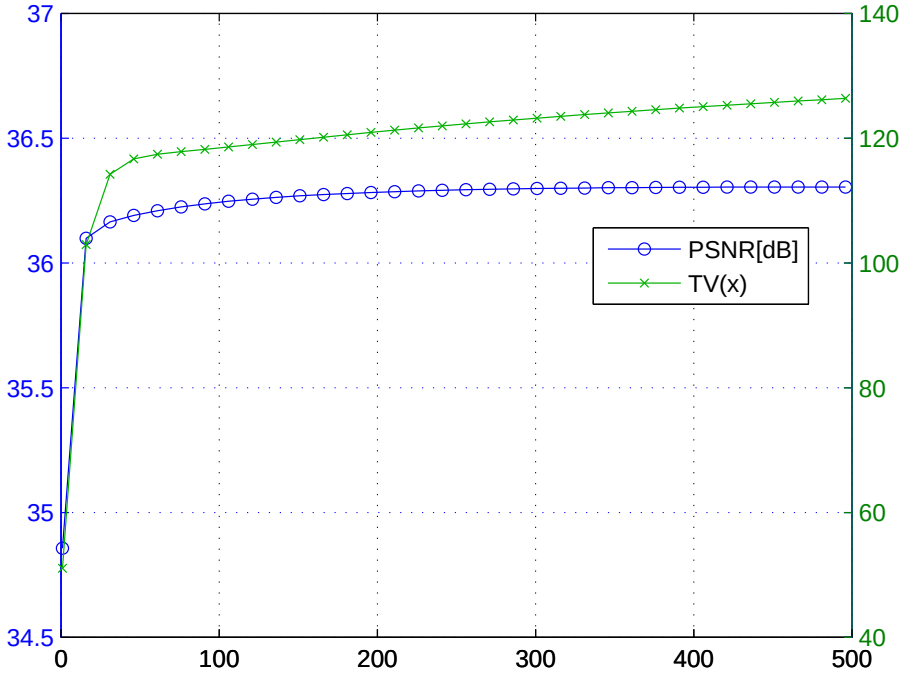


Figure C.25: PSNR of AMID algorithm evolution for Suzie video frame along with TV

Figure C.26: Carphone video: (top-left) original frame; (top-right) degraded; (bottom-left) SAI + BTV deblurring 31.6304dB; (bottom-left) AMID 31.8205dB

Figure C.27: Foreman video: (top-left) original frame; (top-right) degraded; (bottom-left) SAI + BTV deblurring 33.2643dB; (bottom-left) AMID 33.9584dB

Figure C.28: Miss America video: (top-left) original frame; (top-right) degraded; (bottom-left) SAI + BTV deblurring 39.1750dB; (bottom-left) AMID 40.0099dB

Figure C.29: Suzie video: (top-left) original frame; (top-right) degraded; (bottom-left) SAI + BTV deblurring 35.9866dB; (bottom-left) AMID 36.3040dB

C.4 SAI-SR: Soft-Decision Adaptive Interpolation Super-Resolution

The SAI algorithm [59] presented in Section C.1 along with the GISD algorithm [114] provide state-of-the-art results in terms of interpolation accuracy and visual quality. Our approach to the problem of super-resolution will be to extend the SAI algorithm to incorporate information presented in other frames of a video than the reference frame.

A great advantage of the SAI algorithm is its formulation. By breaking the interpolation process in two steps, it transforms an ill-posed problem into two well-posed problems. Specifically, if we wish to double the resolution of an image, say $N_1 \times N_2 = N_1 N_2$ pixels, we need to obtain $(2N_1-1)(2N_2-1) = 4N_1N_2-2N_1-2N_2+1$ pixels, which is almost four times the number of pixels in the original image.

The first step of the SAI algorithm produces $(N_1-1)(N_2-1) = N_1N_2-N_1-N_2+1$ pixels from the original N_1N_2 pixels. Due to the formulation of the problem and since the dimensions are compatible, we do not need a regularization term. That is, the input dimension (N_1N_2) is larger than the output dimension $((N_1-1)(N_2-1))$ and, therefore, we solve a stable least-squares problem.

For the second step, we use the original pixels plus the pixels generated in the first step, which corresponds to $2N_1N_2 - N_1 - N_2 + 1$. From these pixels, we need to generate $2N_1N_2 - N_1 - N_2$ in order to complete the interpolation. Again, the dimensions are compatible and we have another well-posed problem which does not need regularization. For a description of the SAI algorithm refer to Section C.1.

C.4.1 SAI-SR: Extension to Incorporate Information from other Frames

In section C.2, we proposed a change in the SAI algorithm, the Modified SAI, in order to consider the whole image at once. This change enhances, in most cases, the quality of the estimated HR image in terms of PSNR with or without post deblurring.

In this section, we will describe an extension of the Modified SAI algorithm to incorporate information in other frames about the reference image. We will call our algorithm *Soft-decision Adaptive Interpolation Super-Resolution* (SAI-SR).

Suppose we have a sequence of images or a video of K frames given by

$$\mathbf{g}_k, \quad k = 0, 1, \cdots K-1, \tag{C.74}$$

where $\mathbf{g}_0$ is referred to as *reference frame*.

The observation model used is

$$\mathbf{g}_k = \mathbf{RHM}_k\mathbf{f} + \mathbf{v}_k, \tag{C.75}$$

where $\mathbf{g}_k$ are the observed and degraded LR images, $\mathbf{R}$ is a decimation operator, $\mathbf{H}$ is a blurring operator, $\mathbf{M}_k$ are the warping or geometric motion matrices, $\mathbf{f}$ is the original HR image and $\mathbf{v}$ is a noise vector.

As suggested in [115] and [108], in case of pure translational motion, the matrices $\mathbf{H}$ and $\mathbf{M}_k$ are block-circulant and thus they commute [116]. We can write the observation model as

$$\mathbf{g}_k = \mathbf{RM}_k\mathbf{z} + \mathbf{v}_k, \tag{C.76}$$

where $\mathbf{z} = \mathbf{Hf}$ is a blurred version of the HR image. Then, the problem can be broken in two parts: a fusion step, which estimates $\mathbf{z}$ from $\mathbf{g}_k$ and a deblurring step, which estimates $\mathbf{f}$ from $\mathbf{z}$.

Although we aim to develop an algorithm for arbitrary motion (not only translational motion), we will use a similar approach to the one used in [85] and [84]. In the case of arbitrary motion, the separation of super-resolution in two steps is not optimal, since it does not respect the non-commutativity of matrices $\mathbf{H}$ and $\mathbf{M}_k$, and leads to inferior results. However, the separation allows for a much simpler algorithm (both conceptually and implementation-wise), so the sub-optimality is the price to be paid for this simplicity.

In equations (C.28) and (C.29) we perform linear regressions to model a pixel value from its spatial neighbors. We now extend those equations to consider other frames, that is

$$\mathbf{b}_{k,l}(\mathbf{n}) = \underset{\mathbf{b}_{k,l}(\mathbf{n})}{\arg\min} \sum_{\mathbf{q}\in W_4} \left[g_k(\mathbf{n}+\mathbf{q}) - \sum_{\mathbf{m}\in\mathcal{T}_b^H} b_{k,l,\mathbf{m}}(\mathbf{n}) g_l(\mathbf{n}+\mathbf{q}+\mathbf{m}) \right]^2, \tag{C.77}$$

and

$$\mathbf{a}_{k,l}(\mathbf{n}) = \underset{\mathbf{a}_{k,l}(\mathbf{n})}{\arg\min} \sum_{\mathbf{q}\in W_4} \left[g_k(\mathbf{n}+\mathbf{q}) - \sum_{\mathbf{m}\in\mathcal{T}_a^H} a_{k,l,\mathbf{m}}(\mathbf{n}) g_l(\mathbf{n}+\mathbf{q}+\mathbf{m}) \right]^2. \tag{C.78}$$

Specifically, equations (C.77) and (C.78) describe linear regressions that model the pixels of frame k from the pixels of frame l. The estimation error of the n-th pixel given by the parameters $\mathbf{a}$ is

$$e_{k,l}(\mathbf{n}) = \left[g_k(\mathbf{n}) - \sum_{\mathbf{m} \in \mathcal{T}_a^H} a_{k,l,\mathbf{m}}(\mathbf{n}) g_l(\mathbf{n}+\mathbf{m}) \right]. \tag{C.79}$$

Considering all pixels of the image, we can lexicographically order the results to obtain the error vector $\mathbf{e}_{k,l}(\mathbf{n})$.

This approach is reasonable if the relative motion between frames, whether local or global motion, is small, i.e., less then 1 pixel in both vertical and horizontal directions. The idea is that, if we have a slight displacement between frame $k = 0$ and $k = 1$, we could better estimate the pixels in the frame $k = 0$ from their neighbors in the frame $k = 1$, i.e., their temporal neighbors. If the estimation error obtained through the temporal neighborhood is higher than the one from the spatial neighborhood, we can somehow "prefer" the latter.

However, if the relative motion between frames is large, i.e., higher than 1, we must compensate it. The reason is that the SAI algorithm, which our SAI-SR algorithm is based on, only considers a neighborhood of size 1 and therefore large motion have to be neutralized. In other words, we neutralize large motion by estimating it and adding this information to the relation between frames. We describe this procedure next. A similar approach was used in [85].

Suppose that we detect a large motion between the reference frame and frame number k, represented by the vector field $\mathbf{o}_k(\mathbf{n}) = [o_1, o_2]^{\mathrm{T}}$. The motion in the vector field $\mathbf{o}_k(\mathbf{n})$ is only the integer part. Then, the parameters should be estimated by

$$\mathbf{b}_{0,k}(\mathbf{n}) = \underset{\mathbf{b}_{0,k}(\mathbf{n})}{\arg\min} \sum_{\mathbf{q} \in W_4} \left[g_0(\mathbf{n}+\mathbf{q}) - \sum_{\mathbf{m} \in \mathcal{T}_b^H} b_{0,k,\mathbf{m}}(\mathbf{n}) g_k(\mathbf{n}+\mathbf{q}+\mathbf{m}+\mathbf{o}_k(\mathbf{n})) \right]^2, \tag{C.80}$$

and

$$\mathbf{a}_{0,k}(\mathbf{n}) = \underset{\mathbf{a}_{0,k}(\mathbf{n})}{\arg\min} \sum_{\mathbf{q} \in W_4} \left[g_0(\mathbf{n}+\mathbf{q}) - \sum_{\mathbf{m} \in \mathcal{T}_a^H} a_{0,k,\mathbf{m}}(\mathbf{n}) g_k(\mathbf{n}+\mathbf{q}+\mathbf{m}+\mathbf{o}_k(\mathbf{n})) \right]^2. \tag{C.81}$$

Conversely, if we are trying to estimate the pixels of the k-th frame from the pixels of the reference frame, we must use the negative of the motion estimation vector field, i.e.,

$$\mathbf{b}_{k,0}(\mathbf{n}) = \underset{\mathbf{b}_{k,0}(\mathbf{n})}{\arg\min} \sum_{\mathbf{q} \in W_4} \left[g_k(\mathbf{n}+\mathbf{q}) - \sum_{\mathbf{m} \in \mathcal{T}_b^H} b_{k,0,\mathbf{m}}(\mathbf{n}) g_0(\mathbf{n}+\mathbf{q}+\mathbf{m}-\mathbf{o}_k(\mathbf{n})) \right]^2, \tag{C.82}$$

and

$$\mathbf{a}_{k,0}(\mathbf{n}) = \underset{\mathbf{a}_{k,0}(\mathbf{n})}{\arg\min} \sum_{\mathbf{q}\in W_4} \left[g_k(\mathbf{n}+\mathbf{q}) - \sum_{\mathbf{m}\in\mathcal{T}_a^H} a_{k,0,\mathbf{m}}(\mathbf{n}) g_0(\mathbf{n}+\mathbf{q}+\mathbf{m}-\mathbf{o}_k(\mathbf{n})) \right]^2 . \tag{C.83}$$

In our experiments, we used the *optical flow* [117] approach to obtain the motion vectors. The algorithm used was the Hierarchical Lucas-Kanade using pyramids [118, 119]. The algorithm provides motion vectors to each pixel of reference frame, i.e., the most likely position of the pixel in another frame.

Now suppose a reference frame, say $k = 0$. We compute the pixel-wise models for all combinations between the reference frame and the other frames in the video, i.e., $\{k = 0, l = 0, 1, ..., K\}$, $\{k = 1, l = 0\}$, $\{k = 2, l = 0\}$, $\cdots$, $\{k = K, l = 0\}$.

We can rewrite equation (C.33) taking into account the various models obtained from equations (C.77) and (C.78)

$$\hat{\mathbf{f}}_k = \underset{\mathbf{f}}{\arg\min} \left\{ \left\| \mathbf{f} - \mathbf{A}_{0,k}^f \mathbf{g}_k \right\|^2 + \left\| \mathbf{C}^g \mathbf{g}_k - \mathbf{A}_{k,0}^g \mathbf{f} \right\|^2 \right\} \tag{C.84}$$

To construct matrix $\mathbf{A}_{k,0}^g$ we use the parameter vectors $\mathbf{a}_{k,0}^g(n)$. These vectors are obtained from the observed image (superscript g), regard the pixel n and are used to estimate the pixels of the k-th frame from the pixels of the reference frame (superscript $\{k, 0\}$).

On the other hand, we construct matrix $\mathbf{A}_{0,k}^f$ from the parameter vectors $\mathbf{a}_{0,k}^f(n)$. This vector is obtained by averaging the parameter vectors from the neighbors of the n-th pixel (subscript x) and are used to estimate the pixels of the reference frame (0) from the pixels of the k-th frame (superscript $\{0, k\}$).

It is important to note that the construction of the matrices $\mathbf{A}_{k,0}^g$ and $\mathbf{A}_{0,k}^f$ take into account the neutralization of large motion as previously mentioned and shown in (C.80) and (C.81).

Solving equation (C.84) as we did with equation (C.33), we obtain

$$\hat{\mathbf{f}}_k = \left(\mathbf{K}_k^{\mathrm{T}} \mathbf{K}_k\right)^{-1} \mathbf{K}_k^{\mathrm{T}} \mathbf{L}_k \mathbf{g}_k \quad k = 0, ..., K. \tag{C.85}$$

The above equation represents the first pass (see SAI algorithm in section C.1 and C.2) estimate of the HR pixels of the reference frame using information contained in the k-th frame.

C.4.2 Combining Estimates

So far, we have obtained estimates regarding the first pass of SAI algorithm considering not only the spatial neighborhood of each pixel in the reference frame, but also

their spatial-temporal neighborhood. The estimates are represented by $\hat{\mathbf{f}}_k$, where for $k = 0$ we have the result from the Modified SAI algorithm (see Section C.2) and for $k = 1, \cdots, K-1$ we have estimates of the reference frame obtained from information presented in other frames. Here, we address how to combine these various estimates.

In [120], Li proposes a scheme for video processing using mixture models. Each model tries to estimate a pixel value in a reference by a linear combination of neighbor pixels both in space (within the reference frame) and time (other frames). The models are mixed according to their confidence in terms of estimation, that is, the estimation error.

In [84], Protter *et al.* generalized the Nonlocal Means filter to super-resolution. The technique basically consists in combining several images patches from the reference frame and from other frames according to their mean-square errors to the reference image patch.

Based on the spirit of these criteria, we propose a scheme to combine the estimates of equation (C.85) as

$$\hat{\mathbf{f}} = \frac{\sum_{k=0}^{K-1} \hat{\mathbf{f}}_k \circ \exp\left(-\mathbf{e}_k^2\right)}{\sum_{l=0}^{K-1} \exp\left(-\mathbf{e}_l^2\right)}, \tag{C.86}$$

where $\mathbf{e}_k$ is the pixel-wise estimation mean-square error vector of the reference frame from the k-th frame (see equation (C.79)). A more appropriate notation would be $\mathbf{e}_{0,k}$, but we dropped the index 0 for simplicity. The symbol "$\circ$" stands for Hadamard or element-wise product and $\frac{a}{b}$ stands for Hadamard or element-wise division.

Unfortunately, experiments have shown that the estimation error itself is not enough to properly combine the estimates. Occlusions and updates in subsequent frames, which cannot be modeled by relative motion, negatively affect the final results. The problem is that motion estimation algorithms produce motion outputs independently of the relation between frames. Even if one frame contains no information about another, i.e., two completely different frames, motion estimation algorithm will yield motion vectors between them.

It turns out that occlusions and updates can occasionally provide lower estimation error between frames. Thus, the mixture based on equation (C.86) erroneously combine the several estimates, giving more weight to wrong estimates and reducing the quality of final result. We propose two extra weighting factors to contemplate such issue.

The first weighting factor is obtained by analyzing the motion vectors. This term prevents erroneously consideration of eventual occlusions and updates. Suppose that we are estimating motion between the reference frame $k = 0$ and the frame $k = 1$. If two or more motion vector point to the same location in the frame $k = 1$, the motion field close to that location is complex. So is the case where there is a pixel in frame

$k = 1$ to which no motion vector is pointing from the reference frame $k = 0$.

In both cases, the information in frame $k = 1$ is not useful to perform estimation in the reference frame, since there is a change in the image. Therefore, we create a mask vector $\mathbf{m}_k$ to avoid the incorrect use of information. The mask is 0 where a complex motion field was detected, and 1 where there is no occlusion or update, i.e., true motion.

For the second weighting factor, we calculate the radiometric (gray-level) proximity between windows in both frames after neutralizing large motion, using a similar approach from the one in [84]. Additionally, we use small motion (smaller than 1 pixel) obtained from the motion estimation algorithm to shift the windows. The second proposed mixture term is given by

$$d_k(\mathbf{n}) = \|\mathbf{P}(\mathbf{n})\mathbf{g}_0 - \mathbf{O}(\mathbf{n})\mathbf{g}_k\|\,, \tag{C.87}$$

where $\mathbf{P}(\mathbf{n})$ extracts a patch from the reference frame $\mathbf{g}_0$ around the position $\mathbf{n}$ and $\mathbf{O}(\mathbf{n})$ extracts a patch from the frame $\mathbf{g}_k$. Additionally, matrix $\mathbf{O}(\mathbf{n})$ neutralizes large motion and compensates for small motion.

The compensation for small motion is achieved by applying sub-pixel shifts on the patches. The amount of sub-pixel shift was previously estimated through the motion estimation algorithm. This procedure enhances the accuracy of the radiometric distance calculation and experiments have shown increased performance in final results.

We can group the results from equation (C.87) in a vector, resulting in $\mathbf{d}_k$. Finally, taking into account the three weighting factors proposed, we have as the combination formula

$$\hat{\mathbf{f}} = \frac{\sum_{k=0}^{K-1} \hat{\mathbf{f}}_k \circ \exp\left(-\mathbf{e}_k^2\right) \circ \mathbf{m}_k \circ \exp\left(-\mathbf{d}_k^2\right)}{\sum_{l=0}^{K-1} \exp\left(-\mathbf{e}_l^2\right) \circ \mathbf{m}_l \circ \exp\left(-\mathbf{d}_l^2\right)}. \tag{C.88}$$

C.4.3 Results of the SAI-SR Algorithm

As we have done with the Modified SAI algorithm, we applied the SAI-SR to well-known video sequences. The sequences were firstly blurred and decimated and then submitted to the algorithm the results are shown in Tables C.5 and C.6. PSRN evolution is shown in Figures C.30, C.31, C.32 and C.33. Final image estimates are also shown in Figures C.34, C.35 and C.36.

Method\Video	Carphone	Foreman	Miss America	Suzie
Cubic spline	29.6453	29.7510	38.2466	34.4888
SAI	29.8432	30.2352	37.9431	34.4902
SAI-SR 1	29.9672	30.0097	37.9738	34.5941
SAI-SR 2	29.9989	30.0040	37.9894	34.5996
SAI-SR 5	30.0108	29.9208	37.9696	34.6047
SAI-SR 10	29.9900	29.7816	37.9764	34.5579
Max gain of K-frame SAI-SR over 1 frame	0.0436	−0.0057	0.0156	0.0106

Table C.5: PSNR results of SAI-SR algorithm without deblurring

Method\Video	Carphone	Foreman	Miss America	Suzie
Cubic spline	30.1728	30.6514	38.9363	35.0976
SAI	30.5350	31.0396	38.7313	35.2343
SAI-SR 1	30.6120	31.2311	38.5876	35.2690
SAI-SR 2	30.6427	31.2321	38.5916	35.2608
SAI-SR 5	30.6587	31.1836	38.5830	35.2790
SAI-SR 10	30.6399	31.1053	38.5910	35.2579
Max gain of K-frame SAI-SR over 1 frame	0.0467	0.0010	0.0040	0.0100

Table C.6: PSNR results of SAI-SR algorithm with deblurring

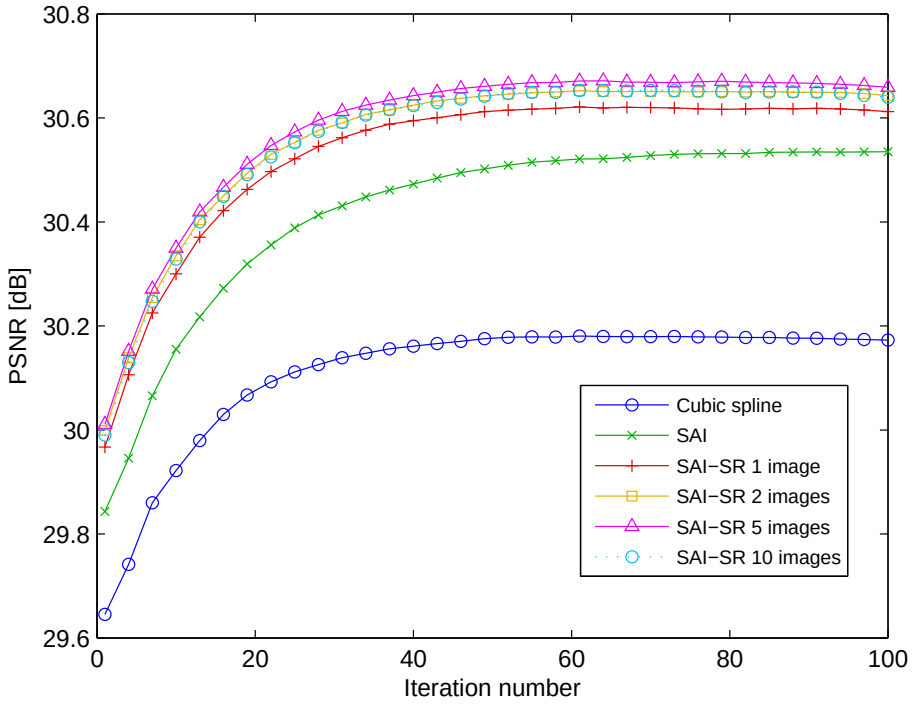


Figure C.30: Deblurring of Carphone video frame via BTV

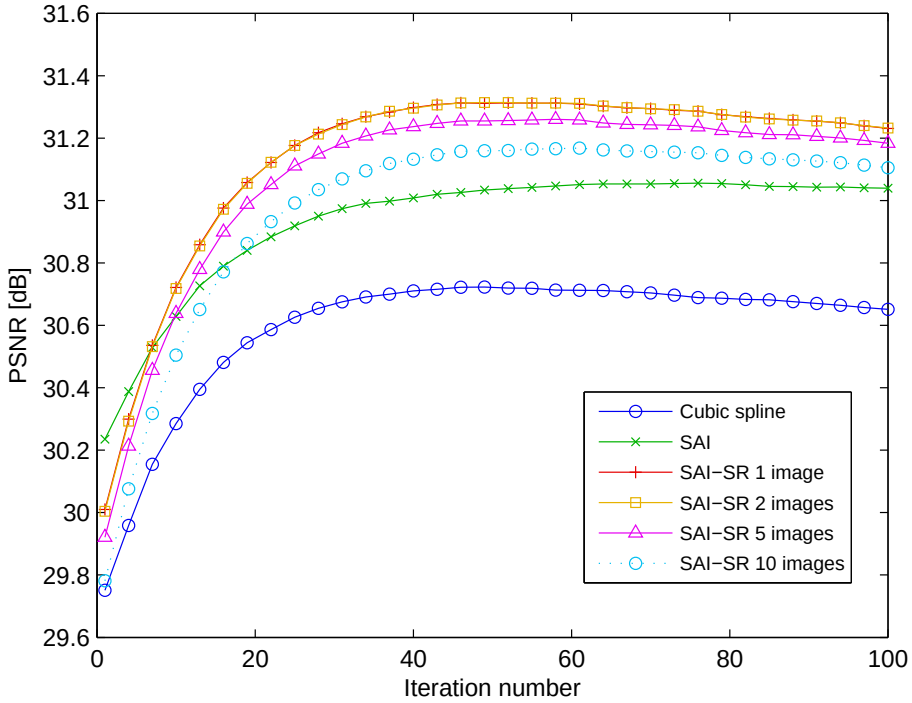


Figure C.31: Deblurring of Foreman video frame via BTV

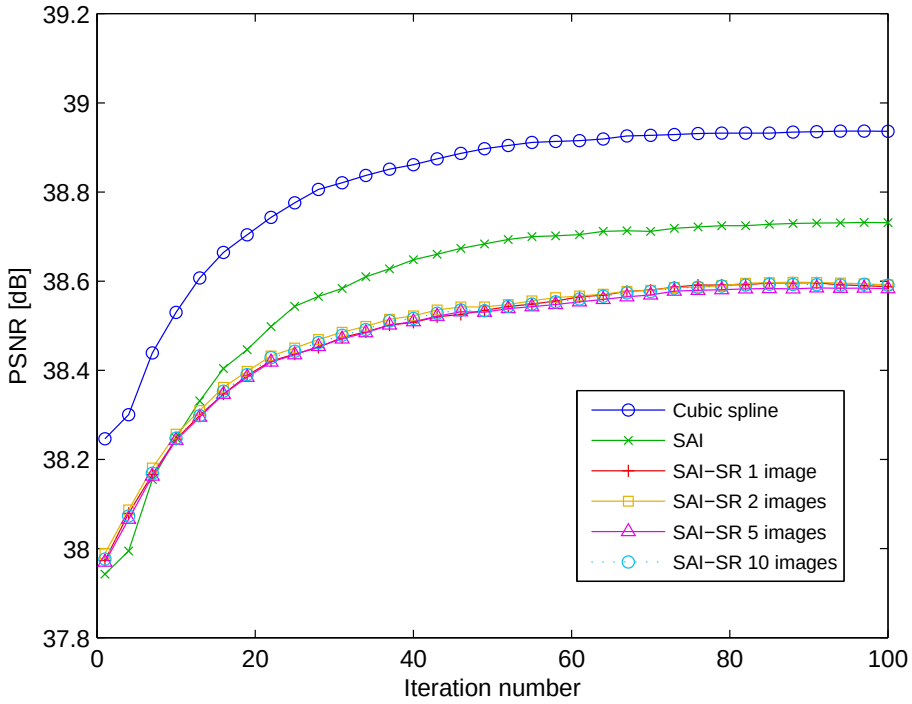


Figure C.32: Deblurring of Miss America video frame via BTV. As with AMID algorithm seen in Figure C.15, both SAI and SAI-SR are worse than cubic spline, probably because the large dark and almost flat area of the image. Imposing geometrical regularity to this area may have led to the failure.

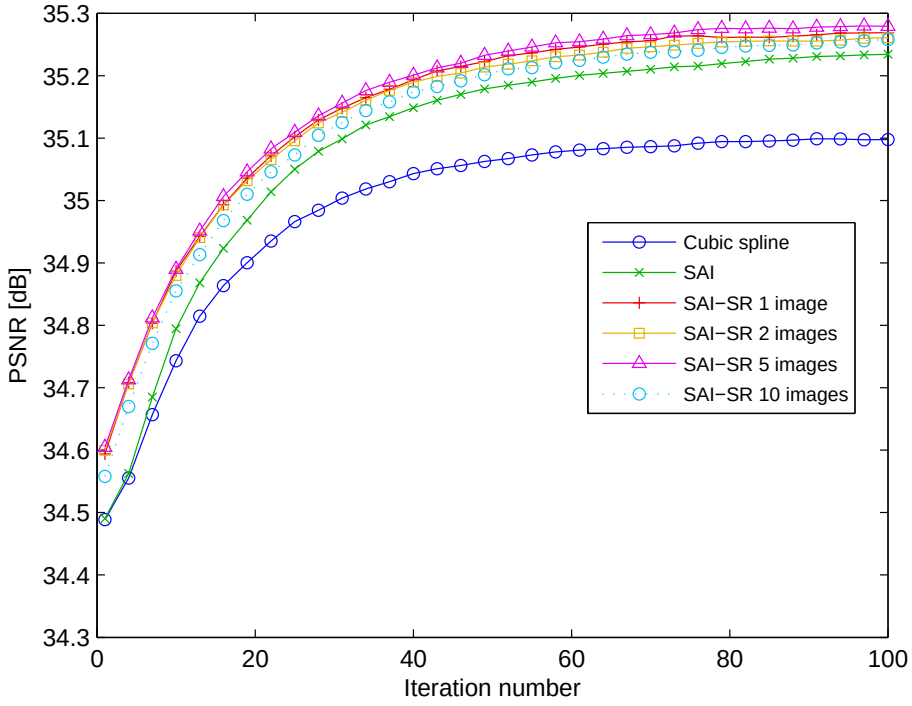


Figure C.33: Deblurring of Suzie video frame via BTV

C.5 Conclusion

AMI and AMID

We first proposed, in Section C.2, a modification in the traditional SAI algorithm [59] in order to consider the whole image at once, rather than separated independent windows. This modification enhanced the PSNR for 3 out of 4 test images. As shown in Figure C.15, however, the results were worse than SAI for Miss America. We believe this occured probably because of the large dark and almost flat area of most of frames. Imposing geometrical regularity to this area may have led to the failure.

Section C.3 presented a new approach to image interpolation and deblurring considering autoregressive modeling. The aim was to perform interpolation and deblurring in one step based on autoregressive models of the image. The motivation was a further extension of the method for performing multi-frame super-resolution in one step.

Before considering the blur, we introduced the Autoregressive Model Interpolation (AMI) algorithm. Starting from an initial interpolated image, the algorithm iteratively refines the estimate. A stop criteria, based on a Total Variation measure, was proposed and showed good agreement with the iteration number of the best estimate. The results for interpolation (as inputs, the images were not blurred before decimation) showed a gain of around 0.3 dB over the traditional SAI method.

Figure C.34: Carphone video: (top-left) original frame; (top-right) degraded; (bottom-left) cubic spline + deblurring 30.1728dB; (bottom-left) SAI-SR 5 images + deblurring 30.6587dB

Figure C.35: Foreman video: (top-left) original frame; (top-right) degraded; (bottom-left) cubic spline + deblurring 30.6514dB; (bottom-left) SAI-SR 2 images + deblurring 31.2321dB

Figure C.36: Suzie video: (top-left) original frame; (top-right) degraded; (bottom-left) cubic spline + deblurring 35.0976dB; (bottom-left) SAI-SR 5 images + deblurring 35.2790dB

Then, we added an extra term to the cost function of the AMI algorithm to take into account the blurring, what resulted in the Autoregressive Model Interpolation and Deblurring method (AMID). The extra term forces the conformity of the estimate and the observed image.

Finally, we presented results of the AMID algorithm. Unlike Total Variation or Bilateral Total Variation deblurring which do not take into account the image characteristics, the AMID algorithm performs a regularized deblurring based on autoregressive models obtained from the observed image. The average gain of AMID over the best setting of parameters for the state-of-the-art SAI and BTV deblurring was 0.5 dB.

SAI-SR

We also presented the development of a new algorithm for super-resolution without the need of sub-pixel motion estimation. Our motivation, aligned with recent developments in this field, is to broaden the applicability of super-resolution algorithms to real videos, specially those containing complex and arbitrary motion. Specifically, early approaches to super-resolution require accurate sub-pixel (HR) motion information, narrowing the usage of the technique.

From the modification of SAI algorithm, presented in Section C.2, we introduced the temporal axis by modeling pixel values in the reference frame from the pixels in other frames. Then, we proposed a mixture formula to combine several estimates of the reference frame.

Section C.4.3 presented some experiments performed with the SAI-SR algorithm and compared the results to other state-of-the-art techniques for image interpolation and deblurring. The results show a coherent behavior, i.e., the image quality in term of PSNR increases when more frames are considered, supporting the *multi-frame* super-resolution concept.

Although the gain of SAI-SR in terms of PSNR from 1 image to K images could be considered marginal (from 0.0010 to 0.0467 dB), it is compatible with the recent exploratory literature in the field. If we consider the overall gain over the Cubic Spline interpolation and BTV deblurring (a very common practice in the literature), the gain becomes around 0.3 dB, which is already visually noticeable.

Wrap-up

Piece-wise autoregressive models provide a powerful and interesting mechanism for regularization in inverse problems. More recent works [56, 57, 121] have been using PAR models in other areas, such as *compressed sensing*.

We can, however, perceive some deficiencies and artifacts in our experiments, notably in Figures C.28 and C.29. We believe that this deficiency can be explained by two reasons:

1. model overfitting, caused by an excessive number of coefficients or by a excessively specific model;
2. a flaw in the model estimation for missing pixels, where we simply averaged the parameters obtained from the neighbors. The failure is clear, for instance, near a corner, where neighboring pixels are very likely to have different models.

■[2]

[2]Conduz-se o leitor de volta ao corpo da tese no Capítulo 5, página 30.

Appendix D

Experiments with PAR models

In order to improve the local description of the image, we carried out some experiments with piecewise autoregressive (PAR) models. Basically, we wish to evaluate how well some estimation strategies perform under the influence of blur and noise.

We calculate the model matrix $\mathbf{A}$ from a blurred and noisy observation $\mathbf{g}$ and compute the residual vector

$$\mathbf{e} = \mathbf{f} - \mathbf{A_g f}, \tag{D.1}$$

where $\mathbf{A_g}$ is the model matrix estimated from the observation $\mathbf{g}$ and $\mathbf{f}$ is the clean image.

D.1 The variables

D.1.1 Model order

The model order $N \times N$ dictates how many neighboring pixels will be used to estimate a certain pixel. Although it is clear that higher orders are able to model a larger class of image features, we want to evaluate if the gain is significant.

Additionally, the more parameters we have to estimate, the more ill-posed becomes the estimation process and usually we have to consider a larger training set.

D.1.2 Training window size

Enlarging the training window $M \times M$ has the effect of increasing the number of equations available and, thus, provides a more stable estimation. When noise level is high, it is expected that larger windows will result in a better model estimation.

On the other hand, larger windows may include regions of the image with different characteristics. Therefore, the training set may not be representative for the feature we are trying to model.

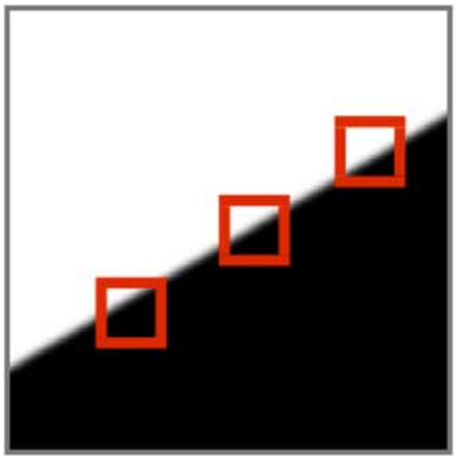


Figure D.1: When training the model for a pixel located on an edge (center of the figure), using a large square window is not representative. A better idea is to use other patches on the same edge for the training set, as shown by the red squares.

D.1.3 Weighted least squares estimation

Figure D.1 illustrates a case where considering a training set distributed along the edge is more representative than using a square window.

We can take this into account by considering a weighted least squares strategy to give more importance to similar patches. Thus, we can enlarge the training window, or more precisely the search window, and give higher weights to pixels that have a similar neighborhood. Similar to [84], we propose to calculate the weights through

$$w(i) = \exp\left\{-\frac{\|\mathbf{p}_0 - \mathbf{p}_i\|_\phi^\phi}{S}\right\}, \tag{D.2}$$

where $\mathbf{p}_0$ is a $P \times P$ patch around the center pixel, $\mathbf{p}_i$ is a patch around the pixel under evaluation, ϕ is the type of norm used and S is a controlling factor.

D.1.4 Elimination of dissimilar patches

Even using weights, we might want to discard pixels which are very dissimilar and keep only L equations. This approach is equivalent to block-matching L patches in a window around the pixel under training.

D.1.5 Summary

Table D.1 summarizes the values of the variables used in the experiment. All the combinations, which can be grouped in a variable vector $\boldsymbol{\theta} = [M; L; P; S; \phi; N]$, are indexed and shown in Section D.1.6. Appendix I shows the images from the Kodak dataset used in the experiment.

Variable	Values
Model order $N \times N$	$N = \{3, 5, 7\}$
Training window size $M \times M$	$M = \{7\}$
Patch size $P \times P$ for WLS	$P = \{3, 5, 7\}$
Correction factor for WLS	$S = \{0.1, 1, 2\}$
Norm used for WLS	$\phi = \{1, 2\}$
Number of equations to be used	$L = \{8, 12, 16, 20\}$
Blur levels	$\sigma_b = \{0, 1.67, 3.33, 5\}$
Noise levels	$\sigma_{\boldsymbol{\nu}} = \{0, 2, 5, 15\}$
Image set	24 images from Kodak dataset

Table D.1: Summary of variables evaluated in the PAR experiment. Number of combinations = 28880.

D.1.6 Indexes for variable vectors

- M: Training window size
- L: Number of patches used in training
- P: Size of patches for WLS
- S: Controlling factor for WLS
- ϕ: Norm for WLS
- N: Model order

index: $[M; L; P; S; \phi; N]$

1[7; 8; 3; 0.1; 2; 3]	23[7; 8; 7; 1.0; 2; 5]	45[7; 12; 5; 2.0; 2; 7]	67[7; 16; 5; 1.0; 2; 3]	89[7; 20; 3; 2.0; 2; 5]
2[7; 8; 3; 0.1; 2; 5]	24[7; 8; 7; 1.0; 2; 7]	46[7; 12; 7; 0.1; 2; 3]	68[7; 16; 5; 1.0; 2; 5]	90[7; 20; 3; 2.0; 2; 7]
3[7; 8; 3; 0.1; 2; 7]	25[7; 8; 7; 2.0; 2; 3]	47[7; 12; 7; 0.1; 2; 5]	69[7; 16; 5; 1.0; 2; 7]	91[7; 20; 5; 0.1; 2; 3]
4[7; 8; 3; 1.0; 2; 3]	26[7; 8; 7; 2.0; 2; 5]	48[7; 12; 7; 0.1; 2; 7]	70[7; 16; 5; 2.0; 2; 3]	92[7; 20; 5; 0.1; 2; 5]
5[7; 8; 3; 1.0; 2; 5]	27[7; 8; 7; 2.0; 2; 7]	49[7; 12; 7; 1.0; 2; 3]	71[7; 16; 5; 2.0; 2; 5]	93[7; 20; 5; 0.1; 2; 7]
6[7; 8; 3; 1.0; 2; 7]	28[7; 12; 3; 0.1; 2; 3]	50[7; 12; 7; 1.0; 2; 5]	72[7; 16; 5; 2.0; 2; 7]	94[7; 20; 5; 1.0; 2; 3]
7[7; 8; 3; 2.0; 2; 3]	29[7; 12; 3; 0.1; 2; 5]	51[7; 12; 7; 1.0; 2; 7]	73[7; 16; 7; 0.1; 2; 3]	95[7; 20; 5; 1.0; 2; 5]
8[7; 8; 3; 2.0; 2; 5]	30[7; 12; 3; 0.1; 2; 7]	52[7; 12; 7; 2.0; 2; 3]	74[7; 16; 7; 0.1; 2; 5]	96[7; 20; 5; 1.0; 2; 7]
9[7; 8; 3; 2.0; 2; 7]	31[7; 12; 3; 1.0; 2; 3]	53[7; 12; 7; 2.0; 2; 5]	75[7; 16; 7; 0.1; 2; 7]	97[7; 20; 5; 2.0; 2; 3]
10[7; 8; 5; 0.1; 2; 3]	32[7; 12; 3; 1.0; 2; 5]	54[7; 12; 7; 2.0; 2; 7]	76[7; 16; 7; 1.0; 2; 3]	98[7; 20; 5; 2.0; 2; 5]
11[7; 8; 5; 0.1; 2; 5]	33[7; 12; 3; 1.0; 2; 7]	55[7; 16; 3; 0.1; 2; 3]	77[7; 16; 7; 1.0; 2; 5]	99[7; 20; 5; 2.0; 2; 7]
12[7; 8; 5; 0.1; 2; 7]	34[7; 12; 3; 2.0; 2; 3]	56[7; 16; 3; 0.1; 2; 5]	78[7; 16; 7; 1.0; 2; 7]	100[7; 20; 7; 0.1; 2; 3]
13[7; 8; 5; 1.0; 2; 3]	35[7; 12; 3; 2.0; 2; 5]	57[7; 16; 3; 0.1; 2; 7]	79[7; 16; 7; 2.0; 2; 3]	101[7; 20; 7; 0.1; 2; 5]
14[7; 8; 5; 1.0; 2; 5]	36[7; 12; 3; 2.0; 2; 7]	58[7; 16; 3; 1.0; 2; 3]	80[7; 16; 7; 2.0; 2; 5]	102[7; 20; 7; 0.1; 2; 7]
15[7; 8; 5; 1.0; 2; 7]	37[7; 12; 5; 0.1; 2; 3]	59[7; 16; 3; 1.0; 2; 5]	81[7; 16; 7; 2.0; 2; 7]	103[7; 20; 7; 1.0; 2; 3]
16[7; 8; 5; 2.0; 2; 3]	38[7; 12; 5; 0.1; 2; 5]	60[7; 16; 3; 1.0; 2; 7]	82[7; 20; 3; 0.1; 2; 3]	104[7; 20; 7; 1.0; 2; 5]
17[7; 8; 5; 2.0; 2; 5]	39[7; 12; 5; 0.1; 2; 7]	61[7; 16; 3; 2.0; 2; 3]	83[7; 20; 3; 0.1; 2; 5]	105[7; 20; 7; 1.0; 2; 7]
18[7; 8; 5; 2.0; 2; 7]	40[7; 12; 5; 1.0; 2; 3]	62[7; 16; 3; 2.0; 2; 5]	84[7; 20; 3; 0.1; 2; 7]	106[7; 20; 7; 2.0; 2; 3]
19[7; 8; 7; 0.1; 2; 3]	41[7; 12; 5; 1.0; 2; 5]	63[7; 16; 3; 2.0; 2; 7]	85[7; 20; 3; 1.0; 2; 3]	107[7; 20; 7; 2.0; 2; 5]
20[7; 8; 7; 0.1; 2; 5]	42[7; 12; 5; 1.0; 2; 7]	64[7; 16; 5; 0.1; 2; 3]	86[7; 20; 3; 1.0; 2; 5]	108[7; 20; 7; 2.0; 2; 7]
21[7; 8; 7; 0.1; 2; 7]	43[7; 12; 5; 2.0; 2; 3]	65[7; 16; 5; 0.1; 2; 5]	87[7; 20; 3; 1.0; 2; 7]	109[7; 8; 3; 0.1; 2; 3]
22[7; 8; 7; 1.0; 2; 3]	44[7; 12; 5; 2.0; 2; 5]	66[7; 16; 5; 0.1; 2; 7]	88[7; 20; 3; 2.0; 2; 3]	

D.2 The measurements

Depending on the noise and blur level of the observed image, we want to find which variable vector yields the best model estimation. In order to evaluate the performance of each of the models obtained, the following measurements were performed on the residual vector $\mathbf{e}$:

1. Mean, which was confirmed to be zero on average and will not be further mentioned.
2. Variance, which is the mean squared error. A model which gives low variance provides a good description of the image.
3. Maximum absolute error, in order to check for outliers. Low maximum absolute error implies that the model is flexible and adaptable to image structures.
4. Shape parameter of a Generalized Gaussian distribution, see Appendix H and next section.

Shape parameter in Generalized Gaussian distribution

From a Bayes perspective, the cost function

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \ \|\mathbf{Hf} - \mathbf{g}\|^2 + \lambda \|(\mathbf{I} - \mathbf{A})\,\mathbf{f}\|^2 , \tag{D.3}$$

which was used earlier in this work, assumes that the residuals are Gaussian. For the term $\|\mathbf{Hf} - \mathbf{g}\|^2$ the assumption is coherent since we assumed an observation model where the additive noise was Gaussian.

The assumption was extended to $\|(\mathbf{I} - \mathbf{A})\,\mathbf{f}\|^2$ in order to simplify the calculations. However, this may not be the case since nothing was assumed for the PAR model.

By determining the shape of the residuals (Gaussian, Laplacian, etc), we can alter the prior norm in the cost function to a more adequate one. For instance, if the residuals are Gaussian distributed, ℓ_2 norm should be used; if they are Laplacian, ℓ_1 norm is more appropriate.

In fact, the average shape parameter of the distribution of residuals was $p = 1.1$ considering all runs. This indicates that ℓ_1 norm should be used for the residuals, i.e. $\|(\mathbf{I} - \mathbf{A})\,\mathbf{f}\|_1$ instead of $\|(\mathbf{I} - \mathbf{A})\,\mathbf{f}\|^2$. A recent work [57] also uses ℓ_1 norm for this term.

D.2.1 Consistency throughout image data set

We also calculated the variance of the measurements within the image dataset. This evaluates the consistency given by a combination of variables.

Blur\Noise	0	2	5	15
0	5	48	48	48
1.67	12	66	48	48
3.33	12	30	30	12
5	12	48	30	12

(a) Minimum variance

Blur\Noise	0	2	5	15
0	5	58	4	58
1.67	12	66	30	37
3.33	12	12	12	30
5	12	48	30	30

(b) Max variance consistency

Table D.2: Evaluation of the variance

Blur\Noise	0	2	5	15
0	31	10	28	19
1.67	12	40	40	1
3.33	12	40	58	37
5	15	4	4	55

(a) Minimum $\left|\hat{\phi} - 1\right|$

Blur\Noise	0	2	5	15
0	16	2	3	3
1.67	2	16	16	16
3.33	6	8	8	3
5	16	6	6	8

(b) Max $\left|\hat{\phi} - 1\right|$ consistency

Table D.3: Evaluation of shape parameter

D.2.2 Summary

Summarizing, we want to find which variable vector produces the minimum variance with maximum consistency for each combination of blur and noise. Additionally, it is desired that the shape parameter be close to 1.

D.3 The results

Table D.2a shows the index of the simulation (see Section D.1.6 for indexes) which resulted in the minimum variance considering all images of the dataset. Table D.2b shows the index of the run which provided the best consistency of this measurement (minimum variance of the measurement between images).

Table D.3a shows the index which provided residuals with shape format closest to 1 and Table D.3b shows the consistency of this measurement. Table D.4 shows the average minimum maximum error.

Table D.5 shows the number of appearances of some indexes in the Tables D.2a, D.2b, D.3a, D.3b and D.4.

D.4 Conclusions

It was not possible to draw a definite conclusion from the experiments, because none of variable vectors satisfied at the same time the desired characteristics, namely:

Blur\Noise	0	2	5	15
0	5	48	66	48
1.67	12	30	48	48
3.33	12	66	30	48
5	12	66	66	12

Table D.4: Minimax absolute error: $\min_i \max_i |[\mathbf{e}]_i|$

Index	Frequency	Variable vector $[M; L; P; S; \phi; N]$
12	20	$[7; 08; 5; 0.1; 2; 7]$
48	13	$[7; 12; 7; 0.1; 2; 7]$
30	12	$[7; 12; 3; 0.1; 2; 7]$
66	7	$[7; 16; 5; 0.1; 2; 7]$
16	5	$[7; 08; 5; 2.0; 2; 3]$

Table D.5: Number of appearances of some indexes as best option in the experiment.

minimum variance, minimum maximum absolute error, maximum consistency and shape parameter closest to 1.

Nevertheless, the experiment provided some insights about model training, which are described next.

1. Shape parameter of the distribution of residuals is on average $\phi = 1.1$.

2. When using WLS, ℓ_2 norm should be used when comparing patches, i.e. $\phi = 2$, which was also used in [84, 122].

3. Although the search window size was fixed at 7×7, discarding pixels with different neighborhood from the training set seems to be a good practice. The ideal total number of pixels L is between 8 and 16.

4. The model order $N \times N$ which yielded the best results was $7 \times 7 = 49$. Obviously, we used a minimum-norm WLS because we had much fewer equations than parameters. This result is reasonable because higher orders provides more flexibility. However, in [56, 57, 59] it is mentioned that models of high order might cause overfitting during reconstruction, fact that was not considered in the experiment.

5. The correction factor S should be set to 0.1, which is around the value suggested in [84].

Rigorously, a more meaningful experiment would be evaluating the average PSNR of reconstructed images using the model estimated with the variables of each index and for each combination of blur and noise.

Later, given a degraded image and estimates of blur and noise levels, one would simply select the best combination of variables for model estimation. In other words, the index which produces the best reconstruction PSRN on average. Clearly, though, the number of runs would dramatically increase and would render the experiment intractable.

Appendix E

Models revisited

A pertinent question is whether *piece-wise autoregressive* (PAR) models are really a good option as *priors* to inverse problems in image processing. We try to answer this question, at least partially, by reviewing some modeling methodologies used in the literature. The focus will be on the use of models in inverse problems and the techniques considered related to PAR models will be described in more detail.

E.1 Analysis and synthesis in inverse problems

Before digging into the models, we highlight and clarify the two approaches that are commonly used in the literature of inverse problems.

E.1.1 Analysis

Although not mentioned before, we have used so far in this work the *analysis* approach, which can be written as

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \ \|\mathbf{W}\mathbf{f} - \mathbf{g}\|^2 + \lambda \|\boldsymbol{\Omega}\mathbf{f}\|_{\phi}^{\phi}, \tag{E.1}$$

where $\boldsymbol{\Omega}$ is referred to as *analysis* operator and ϕ is the norm used in the *prior*.

In the analysis approach we choose, among all possible solutions[1] to the problem, the one that possess the least amount of certain undesired characteristics. Those characteristics are extracted from the image through the analysis operator, i.e., $\boldsymbol{\Omega}\mathbf{f}$.

The analysis approach is quite common in inverse problem, encompassing Tikhonov regularization, Total Variation and many others [1]. In the case of anisotropic TV, for instance, the gradient matrix $\mathbf{D}$ along with the ℓ_1 norm are used as analysis operator. Thus, when TV is used as *prior*, we select the image with the minimum total variation which also maintains fidelity to the observations.

[1] Solutions that are consistent with the observations.

E.1.2 Synthesis

More recent and supposedly stimulated by the *basis pursuit* method [123], the *synthesis* approach initially assumes that the signal $\mathbf{f}$ can be represented by a linear combination of "few" vectors, which are referred to as *atoms* [68]. The atoms would be the basic structures from which the image can be built.

Mathematically, we could construct an image through

$$\mathbf{f} = \sum_i u_i \boldsymbol{\psi}_i \tag{E.2}$$

or, using vector notation,

$$\mathbf{f} = \boldsymbol{\Psi}\mathbf{u}, \tag{E.3}$$

where the coefficients $\{u_i\}$, are often grouped in the vector $\mathbf{u}$. The atoms $\{\boldsymbol{\psi}_i\}$ are usually arranged in columns to form a matrix $\boldsymbol{\Psi}$ which is then called a *dictionary*. If the dictionary has more columns than rows, $\boldsymbol{\Psi}$ is denominated a *redundant dictionary*[2].

If $\boldsymbol{\Psi}$ is sufficiently representative of some signal (and possibly redundant, with many choices of basic structures), we expect to need only few atoms to generate the signal and, therefore, the elements of $\mathbf{u}$ are mostly zero. This concept is termed *sparse representations*.

The key idea is to recover the original image through

$$\hat{\mathbf{f}} = \boldsymbol{\Psi} \arg\min_{\mathbf{u}} \ \|\mathbf{W}\boldsymbol{\Psi}\mathbf{u} - \mathbf{g}\|^2 + \lambda \|\mathbf{u}\|_\phi^\phi, \tag{E.4}$$

where $\phi < 2$ is used to promote sparsity in $\mathbf{u}$, i.e., maximize its number of zero elements.

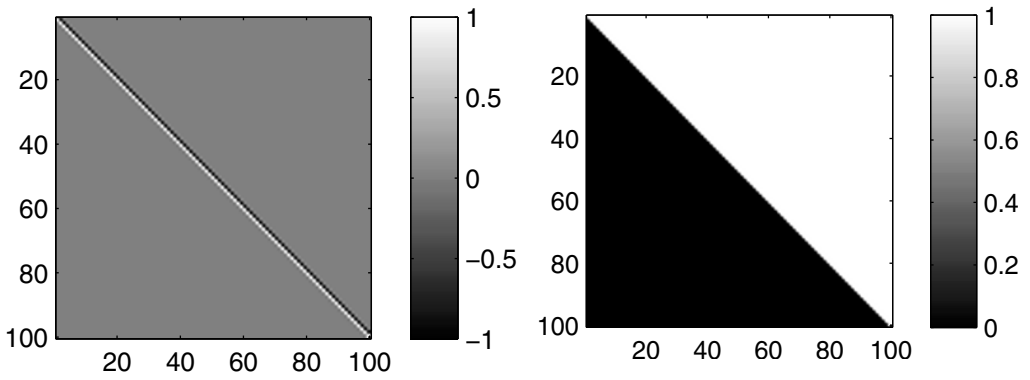
In the synthesis approach we chose, among all the possible solutions to the problem, the one that can be formed with the lowest number of atoms from a certain dictionary. It is clear that this approach is effective only if the initial assumption that $\mathbf{f}$ has a sparse representation in $\boldsymbol{\Psi}$ is valid.

E.1.3 Analysis versus synthesis - I

In the synthesis approach, the focus is on the design of good dictionary, because the final answer will be necessarily in the column space of $\boldsymbol{\Psi}$. Moreover, only few atoms should be used.

On the other hand, the analysis approach focuses on undesired features. Such features will be penalized during the reconstruction, i.e., the solution is not expected

[2]Additionally, the matrix $\boldsymbol{\Psi}$ must be full rank. When it is square and full rank, it is simply called a linear basis. Redundant dictionaries are also called *frames* using linear algebra nomenclature.



(a) Analysis matrix for the 1D TV (b) Synthesis matrix of step functions.

Figure E.1: The matrices in (a) and (b) are inverse. Reconstruction through 1D TV promotes piecewise constant signals. The same signals can be synthesized from shifted step functions (Heaviside) [70].

to possess them.

Despite these fundamental differences, in some cases both approaches are indeed equivalent. Suppose the analysis operator $\mathbf{\Omega}$ is a nonsingular squared matrix. We can define $\mathbf{u} = \mathbf{\Omega f}$ and write $\mathbf{f} = \mathbf{\Omega}^{-1}\mathbf{u}$. Combining this with (E.1) yields

$$\hat{\mathbf{f}} = \mathbf{\Omega}^{-1} \arg\min_{\mathbf{u}} \left\| \mathbf{W}\mathbf{\Omega}^{-1}\mathbf{u} - \mathbf{g} \right\|^2 + \lambda \left\| \mathbf{u} \right\|_{\phi}^{\phi}, \tag{E.5}$$

which is equivalent to (E.4). Specifically, the dictionary is simply the inverse of the analysis operator, i.e., $\mathbf{\Psi} = \mathbf{\Omega}^{-1}$. Figure E.1 shows a classic example of a pair of matrices representing two related analysis and synthesis operators.

For the *undercomplete* case, $\mathbf{\Omega} \in \mathbf{R}^{M \times N}$, $\mathbf{\Psi} \in \mathbf{R}^{N \times M}$, $M < N$, the equivalence still holds and is demonstrated in [68]. Since only few atoms in the dictionary and few "analyzers" in the analysis operator are available, the correspondence is obtained through generalized inverse.

However, for the *overcomplete* case, $M > N$, where dictionaries and analyzers are redundant, the approaches diverge and yield distinct solutions [68]. This subject will be revisited in Section E.5.

E.2 Generative models

The synthesis approach is considered the most intuitive. It is more natural to think of constituent structures of the images (atoms) than non-constituents or forbidden structures ("analyzers"). Therefore, we will initially review generative models that have been traditionally used when solving inverse problems through the synthesis

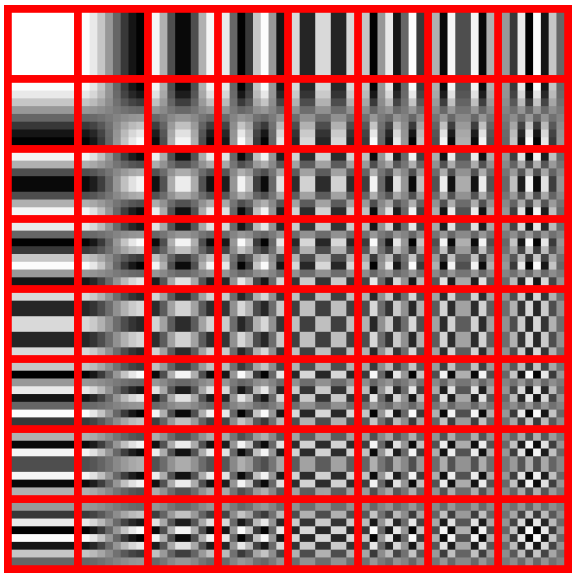


Figure E.2: Atoms of a 2D DCT used in the JPEG image compression format.

approach.

E.2.1 DCT - Discrete cosine transform

When restricted to local small blocks, e.g. 8×8, natural images can be approximated by stationary processes. This implies that cosines are eigenvectors of their autocorrelation matrices. In addition, natural images are highly correlated. These facts explain why the discrete cosine transform (DCT) is a good model natural images. Figure E.2 shows an example of the two-dimensional DCT used in the JPEG format.

The difference between the DCT and the DFT stems from boundary conditions. While the DFT imposes a periodic repetition of the signal beyond its support, the DCT implies in a mirroring, or an even extension. Because images are highly correlated, this mirroring property of DCT produces a greater energy compaction when applied to image blocks, i.e., most of the signal information tends to concentrate in a few low-frequency components.

Because of its interesting properties, DCT has been used, as well as variants such as the redundant DCT [68], in many inverse problems. Examples of algorithms for denoising, deblurring and interpolation using DCT can be found in [68, 70, 124–129].

E.2.2 Wavelets and descendants

Wavelets [130], curvelets [131–133], ridgelets [134, 135] contourlets [136] are some examples of vast family of multi-resolution decompositions that have been extensively applied to image processing problems.

Such transforms provide efficient ways to decompose images. In this case, ef-

ficiency means the ability to represent a signal using just few atoms. Unlike the DCT coefficients, which are concentrated near the origin, wavelet transforms and descendants promote sparsity of the coefficients. Specifically, it is expected that most of the coefficients are zero, though there is no predetermined location in the transform domain.

Using ℓ_0 norm[3], we can solve an inverse problem through

$$\hat{\mathbf{f}} = \mathbf{\Psi} \arg\min_{\mathbf{u}} \ \|\mathbf{W}\mathbf{\Psi}\mathbf{u} - \mathbf{g}\|^2 + \lambda \|\mathbf{u}\|_0 \,. \tag{E.6}$$

A tractable alternative to ℓ_0 norm is the ℓ_1 norm, which also promotes sparsity [137]. Examples of algorithms that use wavelets for solving inverse problems can be found in [70, 138–141].

E.2.3 Redundant dictionaries - I

In order to create a richer and more representative dictionary for images with mixed characteristics, one can combine atoms from different dictionaries. In [142], for example, DCT vectors were used to represent textures, whereas wavelet, curvelet and ridgelet atoms were used to represent smooth areas and contours.

E.2.4 Natural image elements

The methods presented so far employ elements of harmonic analysis that, heuristically, provide good results in image representation. However, one can question whether those elements are indeed the real basic components of images, like pieces of a puzzle from which any natural image could be generated.

This idea has been explored in several works [13, 143, 144], where interesting parallels with mammals' visual system are drawn. The basic supposition is that if animals have evolved to interpret natural images, then their visual systems should be formed by structures specialized to process natural images components.

Analysis on the primary visual system of mammals demonstrated that its structures and cells are characterized by spatial location, orientation and band-pass response. In [143], it was found that decompositions of natural images that promote sparsity also generate atoms with those characteristics. Contrasting to previous approaches that had not yielded results of such significance, the success of [143] was attributed to a greater statistical independence afforded by sparsity.

In [13], similar results were obtained using ICA (independent component analysis) [145]. Although the name of the technique emphasizes the independence of

[3] ℓ_0 is not a norm in the strict sense (it does not satify norm axioms). However, it is often used as a count of nonzero elements of a vector.

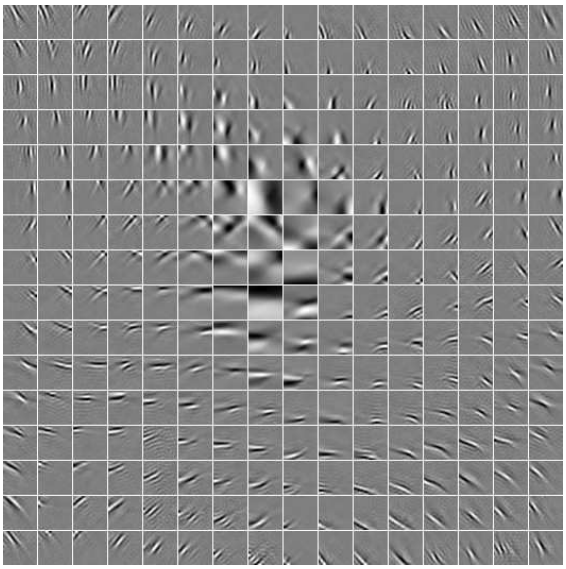


Figure E.3: Example of atoms generated by topographic ICA from a set of 50.000 natural image patches (32×32). The layout of the components is related to their statistical dependence (the closer, the more dependent). Moreover, these components have similar characteristics to the cells of the primary visual system of mammals [13].

components generated, it was observed that the results obtained by ICA were not really independent, and better results were obtained with topographic ICA [146, 147] (see Figure E.3) and ISA (independent subspace analysis) [148–150].

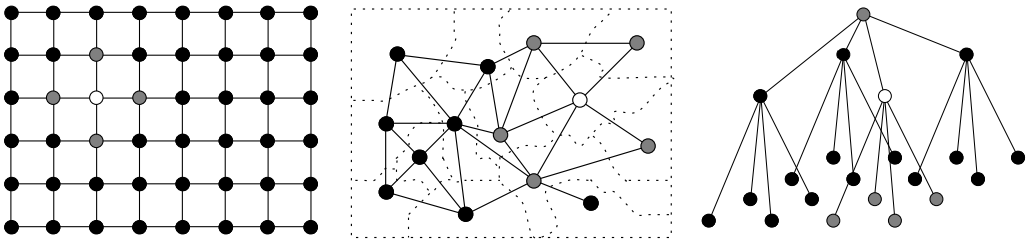
Both approaches share the following idea: the components (atoms) are supposed to belong to different classes or clusters. Then, the optimization is driven towards the maximization of independence between classes, rather than between the atoms themselves.

The atoms of a certain class obtained by this procedure are, in fact, rotated and translated versions of the same feature, i.e., the feature represented by the class. In other words, such approaches yield rotational and translational-invariant representations.

E.2.5 Redundant dictionaries - II

Although pre-constructed dictionaries (from DCT, wavelets and descendants, as seen in Section E.2.3) provide fast implementations, such approaches are limited to the signals for which they were designed and have no direct connection with natural images [70]. An alternative approach is to design redundant dictionaries directly from natural image patches.

A widely used method for this purpose is the K-SVD algorithm [95], which bears resemblance to clustering and vector quantization techniques. K-SVD is a two-step iterative algorithm: given an initial dictionary $\mathbf{\Psi}$, the first step encodes a set of I



(a) (b) (c)

Figure E.4: Three typical graphs of MRF-based models used in image processing [152]. The gray nodes are neighbors of each white node. (a) Rectangular grid of first order. (b) Non-rectangular graph associating partitions of an image. (c) Tree graph.

patches obtaining a set of coefficient vectors $\{\mathbf{u}_i\}_{1\leq i\leq I}$. The second step uses these coefficient vectors to update the dictionary atoms in order to better represent the set of patches. The process is repeated iteratively until some convergence criterion is attained. The equation (E.7) summarizes the procedure.

$$\mathbf{\Psi} = \arg\min_{\mathbf{\Psi},\mathbf{u}} \sum_{i=1}^{I} \left\{ \left\| \mathbf{f}_{(i)} - \mathbf{\Psi}\mathbf{u}_i \right\|^2 + \lambda \left\| \mathbf{u}_i \right\|_0 \right\} \tag{E.7}$$

Examples of inverse problems using dictionaries obtained by K-SVD can be found in [95].

E.3 Energy-based models

Traditionally, when the analysis approach is used in inverse problems, energy-based models are employed. These are inspired in statistical models of mechanical systems. Gray levels and contours of images are linked to states of atoms and molecules in physical systems. We present next some examples of such approaches.

E.3.1 MRF - Markov Random Fields

The theory of *Markov random fields* (MRF) is a branch of probability that studies spatial dependence between physical phenomena. The first use of MRF in image processing was in [151] and description that follows takes this approach.

MRF are based on the theory of undirected graphs. A graph $\mathcal{G} = (\mathcal{S}, \mathcal{N})$ is ordered pair comprising a set of vertices, sites or nodes $\mathcal{S}$ and a set of edges $\mathcal{N}$ linking the nodes, see Figure E.4. Although several analogies are possible, we will use here the correspondence where each node represents a pixel and each edge represents some dependency between two pixels.

Besides being undirected graphs, MRF have the Markovian property, where the conditional probability of future states depends only on the current state. For the case of images, this idea is translated into the assumption that a particular pixel intensity $i \in \mathcal{S}$ depends only on the statistics of its neighbors $i' \in \mathcal{N}_i$ which can be first-order (only adjacent pixels) or higher order dependencies, involving farther pixels.

For regular neighboring systems, we can use the following definition

$$\mathcal{N}_i = \{i' \mid \|\mathbf{n}_i - \mathbf{n}_{i'}\| < r,\ i' \neq i\}, \tag{E.8}$$

where $\mathbf{n}_i$ and $\mathbf{n}_{i'}$ are pixel coordinates and r is the neighborhood radius.

Fixing the neighborhood, we call *cliques*, denoted by c, pixel indexes which are grouped in the sets

$$\mathcal{I}_1 = \{i\} \tag{E.9}$$
$$\mathcal{I}_2 = \{\{i,\, i'\} \mid i' \in \mathcal{N}_i\} \tag{E.10}$$
$$\mathcal{I}_3 = \{\{i,\, i',\ i''\} \mid i',\, i'' \in \mathcal{N}_i\} \tag{E.11}$$
$$\vdots \tag{E.12}$$
$$\mathcal{I}_N = \{\{i,\, i', \cdots, i^{(N)}\} \mid i',\, i'', \cdots i^{(N)} \in \mathcal{N}_i\}, \tag{E.13}$$

which comprise individual pixels, pair of pixels, triples and so on [153]. We call *maximal cliques* $\mathcal{I}_N$ the sets encompassing a certain pixel index i and all its neighbors. From now on, we will use $\mathcal{I}$ for the set of all maximal cliques.

Using MRF, the probability of an image can be given by

$$p(\mathbf{f}) = \frac{1}{Z} \exp\left\{ -\sum_{i \in \mathcal{I}} U(\mathbf{f}_{(i)}) \right\}, \tag{E.14}$$

where $U(\cdot)$ is referred to as *potential function* and $\mathbf{f}_{(i)}$ are the pixels of the maximal clique c or, equivalently, an image patch around the pixel i.

Although MRF can be quite generic, the maximal cliques are commonly restricted to the direct neighbors and the model is simplified to [80, 151]

$$p(\mathbf{f}) = \frac{1}{Z} \exp\left\{ -\sum_{(i,j) \in \mathcal{E}} U(f_i, f_j) \right\}, \tag{E.15}$$

where the potential is usually defined for pixel differences, i.e.,

$$U(f_i, f_j) = \rho(f_i - f_j) \tag{E.16}$$

Figure E.5: Filters that perform first and second derivatives used in [154].

with $\rho(\cdot)$ being some truncated quadratic function or simply $\rho(x) = |x|$. The latter, for instance, can be used to described the total variation (TV) *prior*. In [154], TV was extended to include second derivatives, see Figure E.5.

In short, MRF provides a very general scheme for image modeling. Examples of MRF in image processing can be found in [153, 155–158]. In terms of image denoising and deblurring, we describe next the approach often assumed the most successful among the MRF-based algorithms.

E.3.2 FoE - Field of Experts

The ℓ_1 norm and the gradient operator $\mathbf{D}$ used in the TV *prior* can be considered somewhat arbitrary. Although this pair provides interesting results, the idea of using another type of filter and other norms or potentials have been explored in several works [159–161]. We present here the technique called *Field of Experts* (FoE) developed in [80], which generalizes filters and potentials from a MRF perspective.

The derivative operators $\mathbf{D}_x$ and $\mathbf{D}_y$ are replaced by vectors $\mathbf{j}_c$ over which the image is projected or filtered, i.e., $\mathbf{j}_c^{\mathrm{T}}\mathbf{f}$, and the ℓ_1 norm is replaced by *experts*, which are functions such as

$$\varphi(x;\alpha) = \left(1 + \frac{1}{2}x^2\right)^{-\alpha} \tag{E.17}$$

which is the *Student-t* distribution used in [162] or

$$\varphi(x;\alpha) = \exp\left\{-\alpha\sqrt{1+x^2}\right\}, \tag{E.18}$$

referred to as *Charbonnier expert* [163], a differentiable equivalent of the ℓ_1 norm.

Using FoE, the probability of an image given by

$$p(\mathbf{f}) = \frac{1}{Z}\prod_{i\in\mathcal{I}}\prod_{c=1}^{C}\varphi(\mathbf{j}_c^{\mathrm{T}}\mathbf{f}_{(i)};\boldsymbol{\alpha}_c). \tag{E.19}$$

We note in (E.19) that the image patches are projected on the C vectors $\mathbf{j}_c$ and, afterwards, these inner products are evaluated by the experts. Then, the values returned by the experts are multiplied in the double product. To an image $\mathbf{f}$ is given high probability if the double product returns a value close to 1. FoE also defines procedures to train the filters $\mathbf{j}_c$ and the expert parameters $\boldsymbol{\alpha}_c$ from a large set of

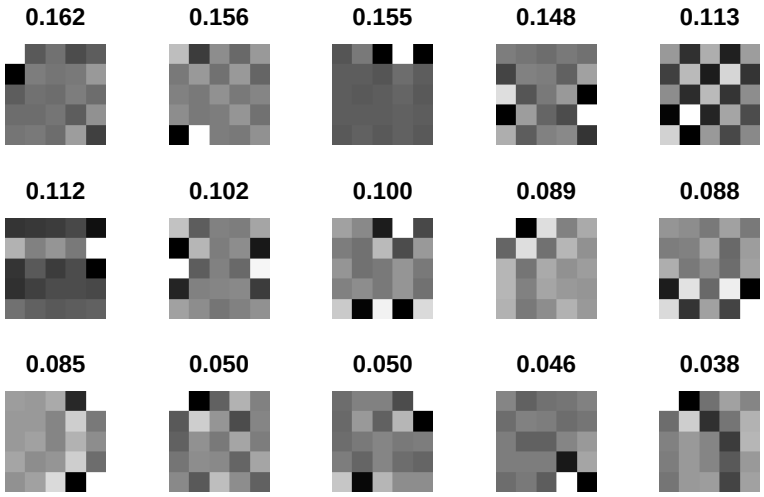


Figure E.6: Example of filters learned with FoE from a set of generic natural images patches with the associated α shown on top [80].

natural image patches.

A very intuitive description of FoE is provided in [164]. First, let's take a look at some filters (Figure E.6) and some experts (Figure E.7) obtained from a training using a generic set of image patches.

The filters are high-pass, although natural images are low-pass. Therefore, the product $\mathbf{j}_c^{\mathrm{T}}\mathbf{f}_{(i)}$ tends to yield low values, which will produce values close to 1 when evaluated by the experts. These, in turn, enter the double product of (E.19) finally assigning high probability to $\mathbf{f}$.

On the other hand, if $\mathbf{f}$ is not a natural image, e.g. noise, the inner products $\mathbf{j}_c^{\mathrm{T}}\mathbf{f}_{(i)}$ are expected to yield high energy which will ultimately assign low probability to the image. In other words, the filters are trained so as they fire rarely on natural images but frequently on all other images. Rather than modeling natural image features, the filters represent "forbidden" characteristics, i.e., features that are not thought as natural [165].

This interpretation is coherent with TV, for example, where we penalize the total variation of the image because we know *a priori* that natural images should have low variation. Similar to the Laplacian distribution used in TV, which "allows" some outliers (the edges of images), the *Student-t* distribution also has long tails. Conversely, though, the parameter α used in FoE provides an extra flexibility since we can fine-tune the importance of each filter in a general description of images.

Interestingly, if we use Gaussian potentials instead of Student-t ones, the optimal filters are given by the latest vectors of the PCA basis of the patches, the so-called *minor components* [166]. The rationale is that the minor components are the least representative of the data and, therefore, should be avoided in the solution. In the FoE framework, the minor components get higher weights (higher values of α),

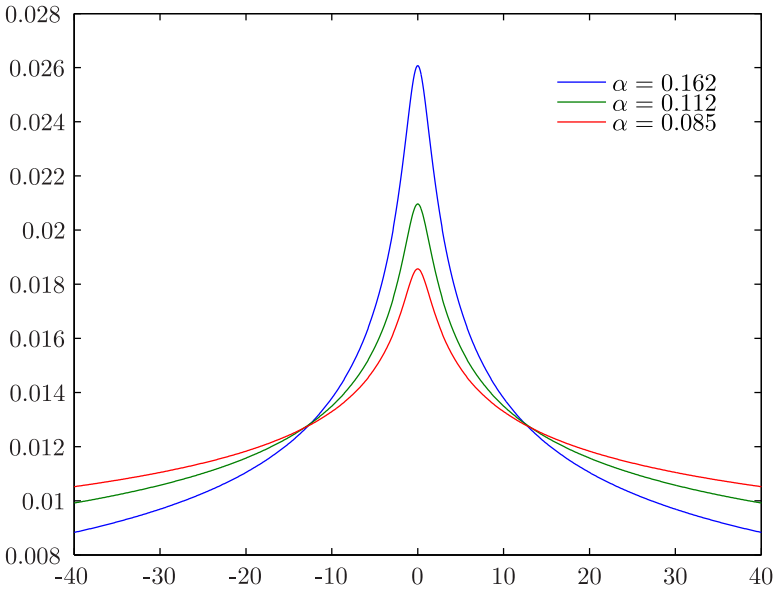


Figure E.7: Example of Student-t distributions for some α corresponding to filters in Figure E.6. The distribution exhibits longer tails for smaller values of α, which indicates lower importance of the correspondent filter. The long tails allow more outliers, signaling a weaker penalization.

meaning that such features should be more severely penalized during reconstruction.

The FoE model developed in [80] uses parametric potentials (Student-t distributions) while the filters are trained from data. A variation developed in [159] adopts fixed filters which are selected from a set of directional derivatives, whereas the potential functions have arbitrary shape and are trained from a set of natural images. However, the approach in [80] generates better reconstruction.

E.4 Nonlocal models

Although parametric models are interesting in terms of versatility, theoretical foundation and generalization ability, the most successful approaches nowadays are those that exploit another aspect of natural images: structures of an image repeat within the same image [167]. In this case, the information that we have *a priori* would not be the structures themselves, but the fact that whatever these structures are, they should appear at different locations. Often, such an approach is referred to as *nonlocal*.

Several methods exploiting this aspect have been proposed lately [84, 122, 168, 169]. We will review here three noteworthy state-of-the-art techniques.

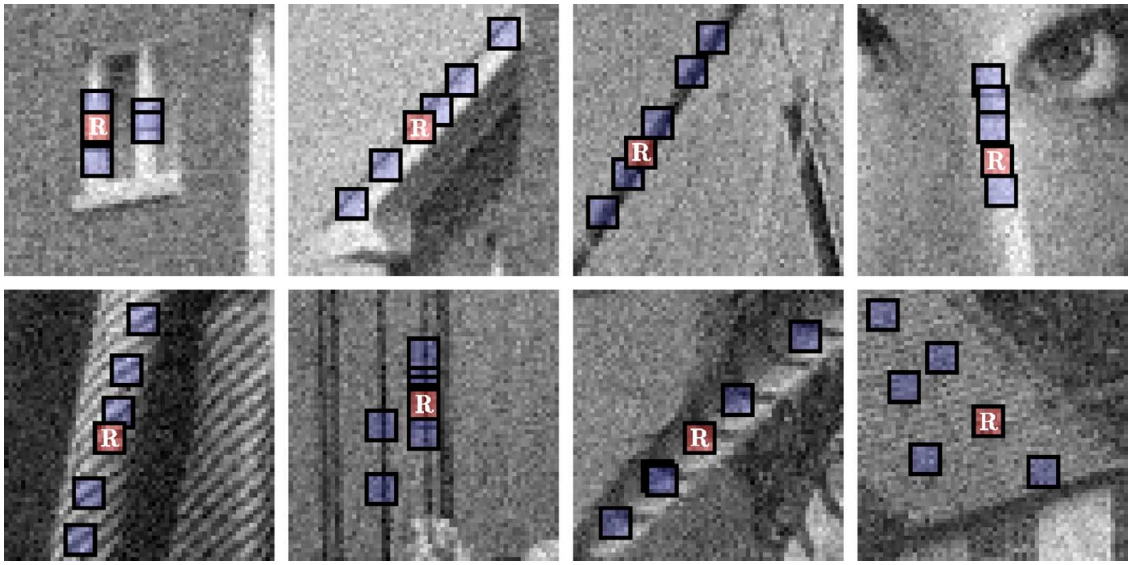


Figure E.8: Block-matching in BM3D. Red squares are reference blocks, whereas blue squares are matched blocks, figure from [81].

E.4.1 BM3D - Block-matching and 3D transform denoising

The algorithm BM3D (*block-matching 3D collaborative filtering*) produces the best results so far in terms of image denoising [33, 81] and will be described below.

First, the image is divided into overlapping blocks. For each block (called reference block) the block-matching technique is used to select similar blocks in a given neighborhood, as illustrated in Figure E.8.

The selected blocks are stacked, along with the reference block, onto a 3D structure, called group. Then, a 3D transform (e.g. 3D DCT, analogous to 2D DCT) is applied to the group in order to decorrelate its components. The resulting coefficients are hard-thresholded and an inverse transform is performed. Finally, the blocks are replaced in their original locations generating a first estimate of the original image, which is called *pilot estimate*. As the blocks overlap each other, a weighted average is employed during the recombination. What has been just described is the first step of the algorithm, which is summarized in Figure E.9.

Using the block indexes from the block-matching in the first step, new 3D structures are formed, but now with blocks from the pilot estimate. Supposing that this new 3D structure is an approximation of the original image, its 3D power spectrum density (PSD) is calculated and then used to design a Wiener filter. The original 3D noisy structures are filtered again, but instead of hard thresholding of transform coefficients, the Wiener filter is used, producing the final estimate. The second step of the algorithm is summarized in Figure E.10.

In a recent work [33], the BM3D method is formalized and analysis and synthesis operators, as described in Section E.1, are derived and used in a decoupled variational

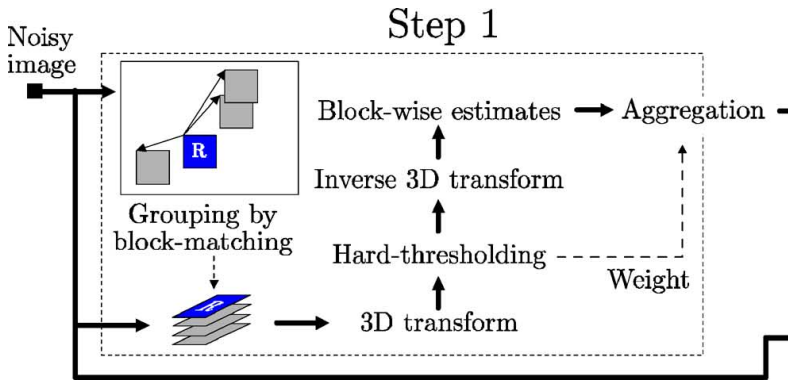


Figure E.9: First pass of the BM3D algorithm, figure from [81].

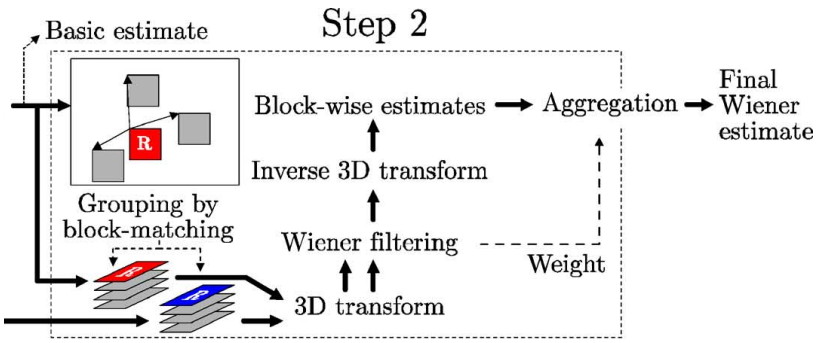


Figure E.10: Second pass of the BM3D algorithm, figure from [81].

scheme for denoising and deblurring.

E.4.2 PLOW - Patch-based locally optimal Wiener denoising

Theoretical limits for image denoising were investigated in [170] using an extended version of the Cramér-Rao lower bound [25]. These limits were compared to different state-of-the-art algorithms and the conclusion was that, for some types of images, there is still room for improvement.

For "complex" images, where structures do not tend to repeat throughout the image (eg non-uniform textures), the results indicate that the existing denoising methods approach the theoretical limit.

For "simple" images, on the other hand, where repetition of structures are common, the study has indicated some room for improvement. Basically, the conclusion was that the paradigm of repetitive structures (nonlocal), though already used in many other methods, seems to have an unexplored potential.

In [96, 171], a denoising algorithm inspired in such theoretical analysis was developed. Basically, it is assumed that the patches that form an image may be categorized into a small number of classes, typically $C = 15$, where each class may represent a particular geometrical structure (corners, contours, flat areas, etc.).

The final estimate of a certain patch is given by the sum of two terms: the first is provided by NLM algorithm (Nonlocal means [122, 168]), which is basically a weighted average of all patches belonging to the same class. However, this method is prone to produce oversmoothed results.

The second term is obtained from the residual between the first term estimate and the original patch. The residual is filtered by Wiener filters, which are designed from the mean and covariance of each class. Finally, the filtered residual is added back to the first term estimate. The rationale is that geometric details that were removed by the weighted average of the NLM algorithm are basically reinserted into the patches.

E.4.3 PLE - Piecewise linear estimation

Another recent work that produces state-of-the-art results in many inverse problems is described in [66]. Using the same idea of patch classification, the algorithm uses piecewise linear estimators (PLE) that actually arise from the assumption that patches can be modeled by Gaussian mixture models. Some previous works that use this technique can be found in [139, 172–174].

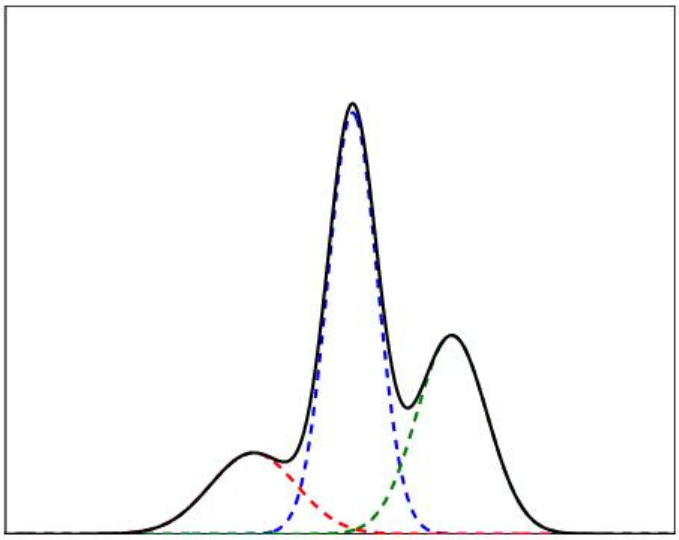


Figure E.11: Illustration of a Gaussian mixture. The color dashed curves represent Gaussian distributions. The black curve is the resultant mixture, figure from [176].

We can define a mixture model by [175]

$$p(\mathbf{f}) = \sum_{c=1}^{C} \alpha_c p(\mathbf{f}; \boldsymbol{\theta}_c), \tag{E.20}$$

where $\{\alpha_c\}$ are the weights of each component and $\{\boldsymbol{\theta}_\mathbf{c}\}$ are the parameters of each distribution. Figure E.11 illustrates this concept.

The method in [66] starts by dividing the image into overlapping patches of $\sqrt{N} \times \sqrt{N}$ (typically 8×8), forming a set $\{\mathbf{f}_{(i)}\}_{1 \leq i \leq I}$. Then, it is assumed that there are C Gaussian distributions $\{\mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)\}_{1 \leq c \leq C}$, typically $C = 18$, parameterized by their means $\boldsymbol{\mu}_c$ and covariance matrices $\boldsymbol{\Sigma}_c$, which together form a Gaussian mixture that describes the patches. It is further assumed that each patch belongs to (or was sampled from) one of these distributions, initially with unknown index $c_i \in [1, C]$, and that the probability of a patch belong to either distribution is uniform.

Mathematically, the probability density function of a patch is, then, given by

$$p(\mathbf{f}_{(i)}) = \frac{1}{(2\pi)^{N/2} \left|\boldsymbol{\Sigma}_{c_i}\right|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{f}_{(i)} - \boldsymbol{\mu}_{c_i})^{\mathrm{T}} \boldsymbol{\Sigma}_{c_i}^{-1} (\mathbf{f}_{(i)} - \boldsymbol{\mu}_{c_i})\right\}, \tag{E.21}$$

provided that the patch was sampled from the distribution (or class) c_i.

As in practice the class is initially unknown, we choose the distribution yielding the highest likelihood, i.e.,

$$\hat{c}_i = \arg\min_c \frac{\left\|\mathbf{W}\mathbf{f}_{(i)} - \mathbf{g}_{(i)}\right\|^2}{\sigma_{\boldsymbol{\nu}}} + (\mathbf{f}_{(i)} - \boldsymbol{\mu}_c)^{\mathrm{T}} \boldsymbol{\Sigma}_c^{-1} (\mathbf{f}_{(i)} - \boldsymbol{\mu}_c). \tag{E.22}$$

Once the most appropriate class $\hat{c}_i$ is identified, the patch estimate is given by

$$\hat{\mathbf{f}}_{(i)} = \arg\min_{\mathbf{f}} \frac{\left\|\mathbf{W}\mathbf{f} - \mathbf{g}_{(i)}\right\|^2}{\sigma_{\boldsymbol{\nu}}} + (\mathbf{f} - \boldsymbol{\mu}_{\hat{c}_i})^{\mathrm{T}} \boldsymbol{\Sigma}_{\hat{c}_i}^{-1} (\mathbf{f} - \boldsymbol{\mu}_{\hat{c}_i}). \tag{E.23}$$

The method proceeds by using the *expectation-maximization* (EM) technique [177], where given an initial estimate of the solution, the model parameters are re-estimated and the process continues alternating solution estimation and model parameter estimation until some convergence criterion is reached.

Let $\mathcal{I}_c = \{i : \hat{c}_i = c\}$ be the set of indexes of all patches belonging to the class c. The class mean and covariance are estimated by

$$\hat{\boldsymbol{\mu}}_c,\, \hat{\boldsymbol{\Sigma}}_c = \arg\max_{\boldsymbol{\mu}_c,\, \boldsymbol{\Sigma}_c} \log p\left(\left\{\mathbf{f}_{(i)}\right\}_{i \in \mathcal{I}_c} \middle|\, \boldsymbol{\mu}_c,\, \boldsymbol{\Sigma}_c\right) \tag{E.24}$$

which are obtained through

$$\hat{\boldsymbol{\mu}}_c = \frac{1}{|\mathcal{I}_c|} \sum_{i \in \mathcal{I}_c} \hat{\mathbf{f}}_{(i)} \tag{E.25}$$

$$\hat{\boldsymbol{\Sigma}}_c = \frac{1}{|\mathcal{I}_c|} \sum_{i \in \mathcal{I}_c} (\hat{\mathbf{f}}_{(i)} - \hat{\boldsymbol{\mu}}_c) \cdot (\hat{\mathbf{f}}_{(i)} - \hat{\boldsymbol{\mu}}_c)^{\mathrm{T}}, \tag{E.26}$$

where $|\mathcal{I}_c|$ is the cardinality of the class c.

Although images are known to be highly non-Gaussian, this study shows that the use of Gaussian mixtures is effective and produces state-of-the-art results in denoising, deblurring and interpolation of images. The key to success, as argued in [66], would be the non-linearity introduced by the selection of best distribution (class) for each patch.

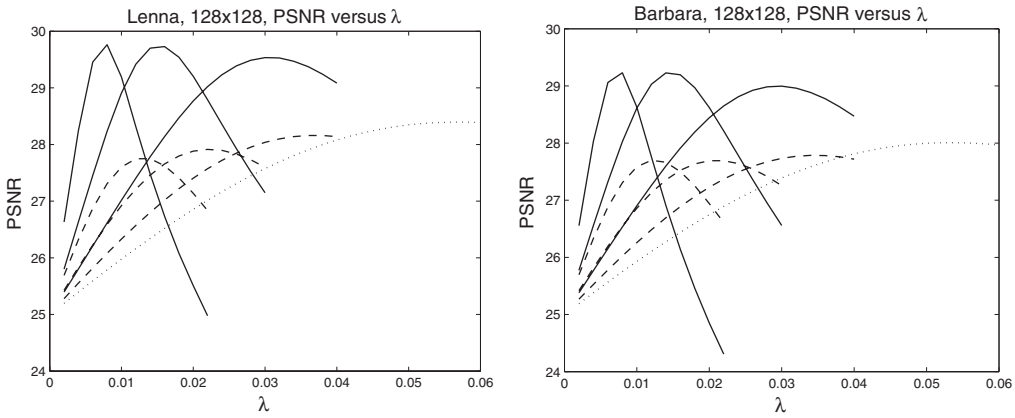
E.5 Discussion

Analysis versus synthesis - II

As mentioned earlier, the analysis and synthesis approaches are not equivalent for the overcomplete case, though they produce the same results for the squared and undercomplete cases. Until the the work by Elad *et al.* in [68], this difference was often considered as a superiority of the synthesis over the analysis.

However, an experiment has revealed the opposite. Using the redundant DCT[4] as *prior* for both cases, the reconstruction PSNR of images using the analysis scheme consistently provided superior results, as shown in Figure E.12. Besides the

[4]The redundancy in the DCT transform is achieved by overlapping the blocks. Depending on how many pixels are shifted, the transform becomes more or less redundant.



(a) (b)

Figure E.12: Analysis versus synthesis in redundant DCT denoising. Solid lines show analysis results. Dashed lines show synthesis results. The three curves (from left to right) correspond to 1, 2 and 3 pixel-shift redundant DCT. Dotted lines show analysis/synthesis PSRN ratio for 8 pixel shift (as block size is 8×8, this means no overlap), figure from [68].

experiment, mathematical support for the claim was also provided in [68]. More empirical evidence can be found in [69, 178]. Recent works [71, 76, 179] have also compared both approaches in different applications. In general, the gap between analysis and synthesis tends to widen up as the redundancy of operators increases.

In sparse representations, the individual importance of each atom is high, since one expects to synthesize signals using few atoms. As argued in [68], if a "wrong" atom is selected in the beginning of a reconstruction process, this could lead to a domino effect, where subsequent "wrong" atoms would keep being chosen to attenuate the first bad choices, resulting in a poor overall performance.

A different argument is that the high overcompleteness in synthesis, rather than positively enriching its descriptiveness, leads to a reverse effect where the dictionary becomes "too descriptive", representing a wide range of undesirable signals. This effect does not apply to analysis where increasing the number of filters still requires the signal to agree with all existing ones [68, 70].

Structured sparsity

As argued in [66], in sparse reconstructions there are too many degrees of freedom, because any atom combination is possible (see Figure E.13a). Since there is no restriction on which atoms can be used, this freedom might lead to an unstable and imprecise estimation [180] due to dictionary coherence[5].

[5]Coherence of a dictionary is defined as the greatest absolute inner product between any two atoms.

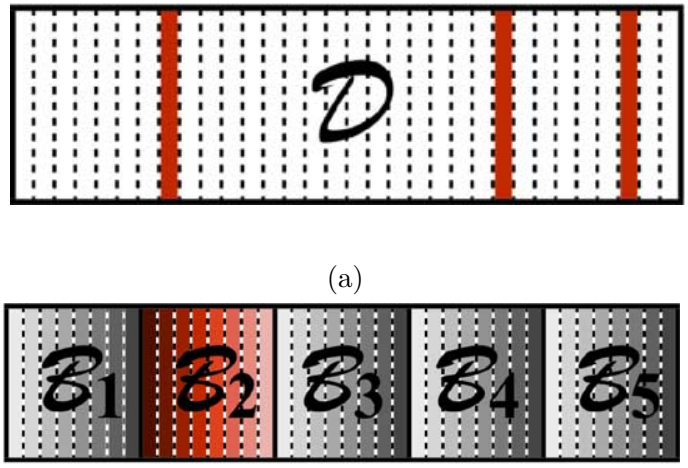


Figure E.13: Redundant dictionary and collection of PCA bases. (a): Redundant dictionary, where each column is an atom. Reconstruction through synthesis has full degree of freedom in choosing atoms (red atoms, for instance). (b) Collection of PCA bases $\mathcal{B}_1, \cdots, \mathcal{B}_5$. Considering all bases, we still have a redundant dictionary, however the reconstruction through the *structured sparsity* approach restrict the allowed atoms to the selected basis (shown in red). This stabilizes the reconstruction. Figure from [66].

Conversely, in the structured sparsity paradigm, the allowed atoms belong to a smaller subgroup, which reduces the degrees of freedom and guarantees a more stable reconstruction. In [66], for instance, the atoms are restricted to the PCA basis which represents the class that was selected as the best Gaussian distribution for the patch, see Figure E.13b.

Nonlocal and collaborative filtering

Also in [66], the concept of structured sparsity is combined to the nonlocal paradigm. Starting from a fixed number of classes, their means and covariances are iteratively adapted to the image content. As the iteration progresses, the classes are more and more customized to the image. All the patches of a certain class profit from this adaptation. This approach, also used in the BM3D method described in Section E.4.1, is often designated as *collaborative filtering*.

Co-sparsity: sparsity after analysis operator

Another kind of sparsity that has been given attention in recent works is the so-called co-sparsity, i.e., sparsity after the analysis operator. Dictionary design methods, such as K-SVD, has also been extended to produce "analysis" atoms, or "analyzers". References can be found in [72, 73, 75, 178, 181–184]. An overview of co-sparsity is

presented in [74].

Nevertheless, we consider that co-sparsity is not a true new concept. It is actually related to previous approaches, such as Field of Experts and Total Variation. The analyzers produced by the analysis version of K-SVD are high-pass filters. FoE and TV also employ high-pass filters and sparsity on the results through the ℓ_1 norm or Student-t distributions.

PAR models versus other models

Piecewise autoregressive (PAR) models can be viewed as a realization of the Markov Random Fields concept. The dependency between pixels are explicitly estimated when PAR coefficients are calculated. PAR models are also related to the PLE (piecewise linear estimators) algorithm, described in Section E.4.3, as will be shown below.

The solution of an inverse problem using PAR models as *prior* can be given by (ℓ_2 norm will be used in the *prior*)

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \ \|\mathbf{W}\mathbf{f} - \mathbf{g}\|^2 + \lambda \|(\mathbf{I} - \mathbf{A})\,\mathbf{f}\|^2 . \tag{E.27}$$

The solution using the PLE algorithm can be written as (setting[6] $\boldsymbol{\mu}_{\hat{c}_i} = \mathbf{0}$ in (E.23) and cleaning up the notation)

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \ \|\mathbf{W}\mathbf{f} - \mathbf{g}\|^2 + \sigma_{\boldsymbol{\nu}} \left\|\boldsymbol{\Sigma}^{-1/2}\mathbf{f}\right\|^2 . \tag{E.28}$$

Setting $\lambda = \sigma_{\boldsymbol{\nu}}$ in (E.27), all we have to show is the connection between $(\mathbf{I} - \mathbf{A})$ and $\boldsymbol{\Sigma}^{-1/2}$ to demonstrate that the approaches can be equivalent. The demonstration that follows has no intention to be rigorous, but only to present intuitive arguments to link the approaches.

Let $\mathbf{f}$ be a natural image patch that can be approximated by a wide-sense stationary (WSS) process with zero mean and covariance matrix $\boldsymbol{\Sigma}$. According to the Wiener–Khintchine theorem [185], $\boldsymbol{\Sigma}$ is diagonalized by the DFT matrix with eigenvalues representing the power spectrum density of the process, i.e.,

$$\mathbf{F}\boldsymbol{\Sigma}\mathbf{F}^{\mathrm{H}} = \mathbf{S} \tag{E.29}$$

where $\mathbf{S} = \mathrm{diag}(s_1^2, \cdots, s_N^2)$ is the eigenvalue matrix.

Multiplication by $\boldsymbol{\Sigma}^{-1/2}$ is equivalent to the inverse filter defined by process

[6] This simplification does not cause loss of generality, as the DC component of the patch can be always reinserted.

spectrum, since

$$\boldsymbol{\Sigma}^{-1/2} = \mathbf{F}^{\mathrm{H}}\mathbf{S}^{-1/2}\mathbf{F} \tag{E.30}$$

with $\mathbf{S}^{-1/2} = \mathrm{diag}(s_1^{-1}, \cdots, s_N^{-1})$. Additionally, if $\mathbf{f} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, then $\boldsymbol{\Sigma}^{-1/2}\mathbf{f} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ [186], that is, $\mathbf{f}$ is decorrelated by $\boldsymbol{\Sigma}^{-1/2}$.

Let $\mathbf{A}$ be a circulant matrix, with zeros in its main diagonal, built from the coefficients of an AR model of $\mathbf{f}$. From $\mathbf{A}$, we can define a transfer function in the z-transform domain by

$$H(z) = \frac{1}{1 - A(z)} \tag{E.31}$$

and the multiplication by $(\mathbf{I} - \mathbf{A})$ is equivalent to

$$H^{-1}(z) = 1 - A(z), \tag{E.32}$$

which is also the inverse filter, demonstrating the connection between the approaches.

One could also show the connection using the Yule-Walker equations [25], which links AR coefficient estimation to the covariance matrix of a WSS process.

E.6 Conclusion

We reviewed in this chapter some common and new approaches to inverse problems in image processing.

We started by highlighting differences and similarities between analysis and synthesis methodologies. We mentioned that, although both approaches are equivalent for the squared and undercomplete cases, they diverge for the overcomplete case.

Noteworthy generative models and methodologies of dictionary design were revised, as well as some important energy-based approaches. More recent nonlocal algorithms were presented as state-of-the-art methods for denoising, deblurring and interpolation.

Finally, new trends and concepts such as the superiority of analysis over synthesis, structured sparsity and collaborative filtering were presented as potential alternatives, and links between PAR models and other approaches were drawn.

■[7]

[7] Conduz-se o leitor de volta ao corpo da tese no Capítulo 6, página 32.

Appendix F

Dealing with fixed-pattern noise

Infrared (IR) focal-plane array (FPA) detectors suffer from fixed-pattern noise (FPN), also known as spatial nonuniformity, which degrades image quality. In fact, FPN remains a serious problem despite recent advances in IRFPA technology. This chapter presents new scene-based correction algorithms to continuously compensate for bias and gain nonuniformity in focal-plane array sensors. The proposed schemes use recursive least square (RLS) and affine projection (AP) techniques that jointly compensate for both bias and gain for each image pixel, presenting rapidly convergence and robustness to noise. Experiments with synthetic and real IRFPA videos have shown that the proposed solutions are competitive with the state-of-the-art in FPN reduction, while presenting recovered images with higher fidelity.

F.1 Introduction

Nowadays, most infrared imaging sensors use Infrared Focal Plane Arrays (IRFPA). Each IRFPA is formed by an array of infrared detectors aligned at the focal plane of the imaging system. Due to the fabrication process, each detector presents unequal responses under the same infrared (IR) stimulus [187]. This spatially nonuniform response produces corrupted images with a fixed-pattern noise (FPN) that has a slow and random drift requiring constant compensation [91]. Hence, the output signal of IR detectors needs to be corrected to produce an image with the quality required by the application. Figure F.1 shows a real-life infrared image corrupted with real FPN.

An accepted approach to FPN correction is to model the pixel responses as affine, that is, a multiplicative term added to a constant [93]; we thus define for each detector (pixel) an offset, or bias, and a gain. By correcting these offsets and gains one aims to obtain a uniform response for the entire FPA. In addition, since these FPA parameters drift over time, such correction has to be performed periodically or even on a frame-by-frame basis.

In most sensors, as the bias nonuniformity dominates the gain nonuniformity, many

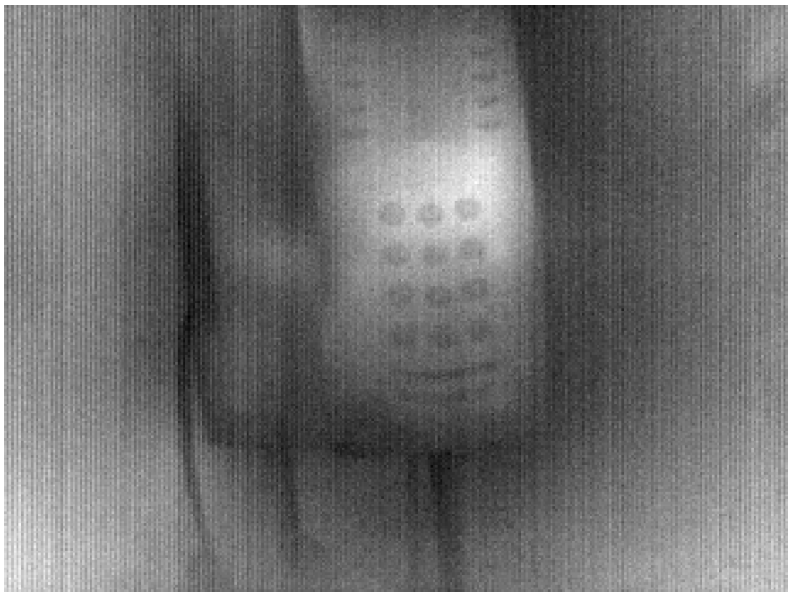


Figure F.1: Image corrupted with real FPN

nonuniformity correction methods do not compensate for the latter [188]. However, better results are achieved when both parameters are corrected. This chapter proposes new adaptive scene-based nonuniformity correction (NUC) algorithms that jointly compensate for bias and gain parameters on a frame-by-frame basis while progressively improving registration. The key contribution of this work is to show how to formulate the bias and gain corrections for NUC using the adaptive filtering framework, particularly those related to the RLS (Recursive Least Squares) and AP (Affine Projection) algorithms. The proposed solutions produce competing reduction in FPN in comparison to the available techniques, while generating perceptually better images.

The rest of the chapter is organized as follows. Section F.2 provides a review of the nonuniformity problem on IRFPA's, as well as the most used correction techniques. Section F.3 is devoted to discuss the NUC techniques pointing out in which class of solution fall the proposed NUC methods. In Section F.4, we briefly provide the signal description. Section F.5 proposes the RLS solution to the NUC which is followed by the corresponding solution employing the AP algorithm in Section F.6. In Section F.7 the experimental results with real and synthetic infrared videos are presented, along with a comparison to other techniques. Section F.9 contains the final remarks and conclusions.

The terms fixed-pattern noise and spatial nonuniformity are used interchangeably.

F.2 IRFPA and fixed-pattern noise models

Although the response of each pixel of an IRFPA is nonlinear, a largely used and accepted model for a FPA sensor is the bias-gain linear model [91, 93, 97, 188–193], given by

$$g_k(n_1, n_2) = a(n_1, n_2) f_k(n_1, n_2) + b(n_1, n_2) \tag{F.1}$$

where $g_k(n_1, n_2)$ is the response (measured signal) of the pixel at position (n_1, n_2) of the IR camera at frame k, $a(n_1, n_2)$ is the gain associated to the (n_1, n_2)-th pixel, $f_k(n_1, n_2)$ is the uncorrupted image, that is, the incident infrared radiation collected by the respective detector at pixel coordinates (n_1, n_2) at frame k, $b(n_1, n_2)$ is the bias associated to the pixel at coordinates (n_1, n_2), and $k = 1, 2, \cdots$ represents the frame number associated to its time instant.

Nonuniformity correction (NUC) algorithms target to estimate the actual infrared radiation $f_k(n_1, n_2)$ by estimating the gain and offset parameters from the readout values $g_k(n_1, n_2)$. Once the bias $\hat{b}(n_1, n_2)$ and gain $\hat{a}(n_1, n_2)$ are estimated, an estimate of the real and uncorrupted infrared image is given by:

$$\hat{f}_k(n_1, n_2) = \frac{g_k(n_1, n_2) - \hat{b}(n_1, n_2)}{\hat{a}(n_1, n_2)}. \tag{F.2}$$

Note that, although bias and gain drift over time, we have dropped their dependency on frame k. This can be done because the drift presented by FPN varies rather slowly. This favors the use of time-invariant parameters modeled together with some tracking of their slow variation.

F.3 Nonuniformity Correction Techniques

If we write equation (F.1) for every pixel (n_1, n_2) and two values of k, we can solve the system of equations and compute $\hat{a}(n_1, n_2)$ and $\hat{b}(n_1, n_2)$, as shown in equation (F.2). However, this solution requires the knowledge of $f_k(n_1, n_2)$. The NUC methods can be categorized in two classes according to the way the values of $a(n_1, n_2)$ and $b(n_1, n_2)$ are estimated: calibration-based (or reference-based), and scene-based.

Reference-based calibration methods for NUC use uniform infrared sources (blackbody radiators) so that $f_k(n_1, n_2)$ is precisely known for all (n_1, n_2). The most widespread technique is the Two-Point Calibration method [93], which employs two blackbody radiation sources at different temperatures to calculate both gain and bias parameters. Despite providing radiometrically accurate corrected imagery, such kind of method interrupts the normal operation of the system during the calibration stage, which is inconvenient in many applications.

Scene-based NUC techniques can overcome this drawback by exploiting motion-

related features in IR videos in order to estimate the gain and bias. In general, these techniques are classified as statistical and registration-based. Registration-based techniques track pixel (or pixel-block) motion between frames, and calculate the associated parameters for the detectors related to the estimated displacements.

Statistical algorithms rely on the assumption that all detectors in the array are exposed to the same range of irradiance (i.e. same statistics) within a sequence of frames. This assumption is valid only if the scene content does not vary significantly from frame to frame. Correction is achieved by adjusting gain and bias parameters of each pixel in order to obtain the same mean and variance for every pixel in the array. Statistical algorithms have been reported by Harris [194], Hayat [195], Torres [189, 190], Scribner [196] and others.

Registration-based algorithms use the idea that each detector should have an identical response when observing the same scene point over time (i.e. same radiance). These algorithms often need a motion-estimation stage to align consecutive frames and compare the responses of two different pixels to the same radiance. Bias and gain are estimated so as the responses become similar. In this case, it is also assumed that the scene does not change considerably between consecutive frames. Registration-based algorithms have been proposed by Sakoglu et al. [193], Hardie [188, 191], Ratliff [91, 197], Averbuch [192], and others. Our methods differ from the previously proposed methods, e.g. Sakoglu et al. [193], because we consider both gain and bias jointly and use a more flexible and general motion model.

A RLS NUC method was presented in [198] by Torres et al. As they point out, the validity of the method is based on the assumption that the scene is constantly moving with respect to the detector, which may not always be true.

Differently, our RLS method assumes only global motion and does not make any assumption on how it varies. Rather, we estimate motion from the frames and use it explicitly when defining the error. Thus, our method can handle a wider class of IR videos.

F.4 Problem statement

As previously mentioned, the key idea is to estimate the bias and the gain associated to each pixel in the image, and then use equation (F.2) to estimate the real and uncorrupted image. First, we write equation (F.1) in vector notation as

$$\mathbf{g}_k = \mathbf{A}\mathbf{f}_k + \mathbf{b}, \tag{F.3}$$

where $\mathbf{g}_k$ is an N-dimensional vector representing the observed image at time k, $\mathbf{A}$ is an $N \times N$ diagonal matrix whose elements are the gain factors associated to the

image pixels, $\mathbf{f}_k$ is an N-dimensional vector representing the real image at time k and $\mathbf{b}$ is a vector representing the bias of the acquired data, with all vectors in a lexicographical order. N is the number of pixels in the image. The gain and the bias (offset) factors are considered time invariant due to their slow drift [190].

If $\hat{\mathbf{A}}$ and $\hat{\mathbf{b}}$ are estimated values of the gain and bias, respectively, an estimation of the real image is given by:

$$\hat{\mathbf{f}}_k = \hat{\mathbf{A}}^{-1}\left(\mathbf{g}_k - \hat{\mathbf{b}}\right). \tag{F.4}$$

As this work proposes the estimation of the bias and gain parameters continuously, we model the variation of the frames in time using a motion equation between two consecutive frames of an IR image sequence as follows:

$$\mathbf{f}_k = \mathbf{M}_k\mathbf{f}_{k-1} + \boldsymbol{\nu}_k, \tag{F.5}$$

where $\mathbf{M}_k$ is the matrix that implements the displacement between consecutive frames $k-1$ and k, and $\boldsymbol{\nu}_k$ is the vector that models the next frame updates that cannot be obtained by a simple displacement.

We suppose that the motion between two successive frames as being obtained by a motion estimation algorithm, and also that vector $\boldsymbol{\nu}_k$ containing the updates is negligible. In this work we perform motion estimation using the LIPSE algorithm described in Section F.8. For more detailed information on the LIPSE algorithm, the reader is referred to [91].

By combining equations (F.3), (F.4) and (F.5), it is possible to write the estimation error vector of frame k based on frame $k-1$, the shift matrix $\mathbf{M}_k$, gain and bias estimates as:

$$\begin{aligned}\boldsymbol{\epsilon}_k &= \mathbf{g}_k - \hat{\mathbf{g}}_k \\ &= \mathbf{g}_k - \hat{\mathbf{A}}\mathbf{M}_k\hat{\mathbf{A}}^{-1}\left(\mathbf{g}_{k-1} - \hat{\mathbf{b}}\right) - \hat{\mathbf{b}},\end{aligned} \tag{F.6}$$

where $\boldsymbol{\epsilon}_k$ is the estimation error vector. The mean square error is given by

$$\varepsilon_k = \frac{1}{N}\sum_{i=1}^{N}\left[\epsilon_k(i)\right]^2 = \frac{\boldsymbol{\epsilon}_k^{\mathrm{T}}\boldsymbol{\epsilon}_k}{N}, \tag{F.7}$$

where N is the total number of pixels in the image.

F.5 RLS algorithm

RLS algorithms aim to minimize a weighted sum of square errors [199, 200], that is

$$\xi_k^{RLS} = \sum_{l=0}^{k} \lambda^{k-l} \varepsilon_l, \tag{F.8}$$

where $0 \ll \lambda \leq 1$ is referred to as *forgetting factor*.

After some manipulation, it can be shown that the update equation for the RLS algorithm may be written as [199, 201]:

$$\hat{\mathbf{b}}_{k+1} = \hat{\mathbf{b}}_k - \hat{\mathbf{H}}_k^{-1} \nabla_{\mathbf{b}} \varepsilon_k, \tag{F.9}$$

where $\hat{\mathbf{H}}_k$ is an estimate of the Hessian matrix and $\nabla_{\mathbf{b}} \varepsilon_k$ is the *a priori* error gradient. The following relations hold for the Hessian matrix [202]:

$$\mathbf{H}_k \triangleq \nabla_{\mathbf{b}}^2 \xi_k^{RLS} = \frac{\partial^2 \xi_k^{RLS}}{\partial \mathbf{b} \partial \mathbf{b}^{\mathrm{T}}} = \lambda \hat{\mathbf{H}}_{k-1} + \frac{\partial^2 \varepsilon_k}{\partial \mathbf{b} \partial \mathbf{b}^{\mathrm{T}}}. \tag{F.10}$$

The above equations show how to update the Hessian matrix at each step.

F.5.1 Bias correction by RLS method

It can be shown that the term $\nabla_{\mathbf{b}} \varepsilon_k$ in equation (F.9) is given by (k index will be hidden for simplicity)

$$\nabla_{\mathbf{b}} \varepsilon = \frac{2}{N} \frac{\partial \boldsymbol{\epsilon}}{\partial \mathbf{b}} \boldsymbol{\epsilon}. \tag{F.11}$$

With this definition, we have that:

$$\frac{\partial \boldsymbol{\epsilon}}{\partial \mathbf{b}} = \left(\mathbf{A} \mathbf{M} \mathbf{A}^{-1} \right)^{\mathrm{T}} - \mathbf{I}. \tag{F.12}$$

The last term of equation (F.10) can be computed as:

$$\begin{aligned} \frac{\partial^2 \varepsilon}{\partial \mathbf{b} \partial \mathbf{b}^{\mathrm{T}}} &= \frac{1}{N} \sum_{i=1}^{N} \frac{\partial^2}{\partial \mathbf{b} \partial \mathbf{b}^{\mathrm{T}}} \left[\epsilon(i) \right]^2 \\ &= \frac{2}{N} \frac{\partial \boldsymbol{\epsilon}}{\partial \mathbf{b}} \frac{\partial \boldsymbol{\epsilon}}{\partial \mathbf{b}^{\mathrm{T}}} \end{aligned} \tag{F.13}$$

Since the term $\frac{2}{N}$ is constant, we define $\hat{\mathbf{H}} = \frac{2}{N} \hat{\mathbf{H}}'$. The complete RLS algorithm for bias correction is given by Table F.1.

The BCGSTABL symbol from the algorithm in Table F.1 represents the solution of the equation $\hat{\mathbf{H}}'_k \mathbf{v}_k = \mathbf{u}_k$ by the so-called BiCGstab(ℓ) – Biconjugate Gradient Stabilized (ℓ) Method [78] in order to avoid the inversion of matrix $\hat{\mathbf{H}}'_k$. This method

Do for $k \geq 0$

$\boldsymbol{\epsilon}_k = \mathbf{g}_k - \mathbf{A}\mathbf{M}_k\mathbf{A}^{-1}\left(\mathbf{g}_{k-1} - \hat{\mathbf{b}}_k\right) - \hat{\mathbf{b}}_k$

$\mathbf{R}_k = \left[\left(\mathbf{A}\mathbf{M}_k\mathbf{A}^{-1}\right)^{\mathrm{T}} - \mathbf{I}\right] \cdot \left[\left(\mathbf{A}\mathbf{M}_k\mathbf{A}^{-1}\right) - \mathbf{I}\right]$

$\hat{\mathbf{H}}'_k = \lambda\hat{\mathbf{H}}'_{k-1} + \mathbf{R}_k$

$\mathbf{u}_k = \left[\left(\mathbf{A}\mathbf{M}_k\mathbf{A}^{-1}\right)^{\mathrm{T}} - \mathbf{I}\right]\boldsymbol{\epsilon}_k$

$\mathbf{v}_k = \mathrm{BCGSTABL}(\hat{\mathbf{H}}'_k, \mathbf{u}_k)$

$\hat{\mathbf{b}}_{k+1} = \hat{\mathbf{b}}_k - \mathbf{v}_k$

Table F.1: RLS algorithm for bias correction

is widely used for solving large sparse unsymmetric linear systems and has provided good results in our experiments.

F.5.2 Why BCGSTABL instead of matrix inversion lemma?

Adaptive filtering literature [199, 200] suggests, for the computation of the Hessian inverse, the use of the matrix inversion lemma

$$[\mathbf{A} + \mathbf{BCD}]^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}\left[\mathbf{D}\mathbf{A}^{-1}\mathbf{B} + \mathbf{C}^{-1}\right]^{-1}\mathbf{D}\mathbf{A}^{-1}, \tag{F.14}$$

where $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ are matrices of appropriate dimensions, and $\mathbf{A}$ and $\mathbf{C}$ are nonsingular. Through this lemma it is possible to update $(\hat{\mathbf{H}}'_k)^{-1}$ with $O(N^2)$ multiplications instead of $O(N^3)$ multiplications needed for direct inversion of $\hat{\mathbf{H}}'_k$.

However, this reduction in complexity is achieved when $\mathbf{B}$ and $\mathbf{D}$ are chosen to be vectors, i.e. $\mathbf{B} = \mathbf{D}^{\mathrm{T}} = \mathbf{f}$ and $\mathbf{C} = \alpha$ is chosen to be a scalar. In this case, the middle term $[\mathbf{D}\mathbf{A}^{-1}\mathbf{B} + \mathbf{C}^{-1}]^{-1}$ is easily inverted as it becomes a scalar, that is, $\left[\mathbf{f}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{f} + \alpha^{-1}\right]^{-1}$.

In our case, $\mathbf{B}$ and $\mathbf{D}$ have to be matrices, i.e. $\mathbf{B} = \mathbf{D}^{\mathrm{T}} = \left[\left(\mathbf{A}\mathbf{M}_k\mathbf{A}^{-1}\right)^{\mathrm{T}} - \mathbf{I}\right]$ and $\mathbf{C} = \mathbf{I}$. Thus, the middle term $[\mathbf{D}\mathbf{A}^{-1}\mathbf{B} + \mathbf{C}^{-1}]^{-1}$ in equation (F.14) is a matrix, which has to be inverted. Therefore, in this case, no complexity reduction is obtained. For this reason, we have chosen to use the BCGSTABL algorithm to avoid matrix inversion as it is suitable to solve large sparse unsymmetric linear systems and tends to converge in few steps (less than 30 iterations).

F.5.3 Gain correction by Tensorial-RLS method

If we apply here a similar procedure used for bias correction in subsection F.5.1, we can write the update equation as

$$\mathbf{A}_{k+1} = \mathbf{A}_k - \hat{\mathbf{G}}_k^{-1}\nabla_{\mathbf{A}}\varepsilon_k, \tag{F.15}$$

where $\hat{\mathbf{G}}_k$ is a Hessian matrix estimate and $\nabla_{\mathbf{A}}\varepsilon_k$ is the *a priori* error gradient[199]. As in equation (F.11), the error gradient is given by (k index hidden)

$$\nabla_{\mathbf{A}}\varepsilon = \frac{2}{N}\frac{\partial \boldsymbol{\epsilon}}{\partial \mathbf{A}}\boldsymbol{\epsilon}. \tag{F.16}$$

The last equation needs the evaluation of the derivative of vector $\boldsymbol{\epsilon}$ with respect to matrix $\mathbf{A}$. This operation is achieved by differentiating each element of vector $\boldsymbol{\epsilon}$ with respect to the full matrix $\mathbf{A}$ [202]. The result is a row-vector in which each element is a matrix, that is, a 3-dimensional tensor. As a result, we refer to the algorithm that we have developed to solve this problem as Tensorial-RLS algorithm.

An approach to the development of these algorithms would be to use tensorial notation and define tensorial operations. Instead, in this work we opted to develop the gain correction by firstly deducing a pixel-by-pixel gain estimator. Then, in section F.5.3, we further develop the method by grouping pixel-by-pixel operations in vectors and matrices, resulting in a compact representation algorithm which is more easily implemented.

The update equation for gain estimation of each pixel by Tensorial-RLS can be written as

$$\hat{a}_{k+1}(i) = \hat{a}_k(i) - \hat{\mathbf{G}}_k^{-1}\nabla_{a(i)}\varepsilon_k, \tag{F.17}$$

where $\hat{\mathbf{G}}_k$ is a Hessian matrix estimate and $\nabla_{a(i)}\varepsilon_k$ is the *a priori* error gradient[199]. The gains associated to the pixels of the image are lexicographically ordered and their individual values are accessed through index i (that is, $a(i) = [\mathbf{A}]_{ii}$).

Strictly speaking, $\nabla_{a(i)}\varepsilon_k$ is a scalar, as it is defined as $\frac{\partial \varepsilon_k}{\partial a(i)}$. Consequently, $\hat{\mathbf{G}}_k$ is not a matrix, but a simple scalar which represents the second-order partial derivative of ε_k with respect to $a(i)$. Thus, the symbol $g_k(i)$ will be used instead of $\hat{\mathbf{G}}_k$.

When the gradient is applied to the error defined in equation (F.7), one gets the following (k index avoided for simplicity):

$$\begin{aligned}\nabla_{a(i)}\varepsilon &= \frac{\partial \boldsymbol{\epsilon}}{\partial a(i)}\boldsymbol{\epsilon} \\ &= -\mathbf{z}^{\mathrm{T}}\left[\frac{\partial \mathbf{A}}{\partial a(i)}\mathbf{M}\mathbf{A}^{-1} + \mathbf{A}\mathbf{M}\frac{\partial \mathbf{A}^{-1}}{\partial a(i)}\right]^{\mathrm{T}}\boldsymbol{\epsilon},\end{aligned} \tag{F.18}$$

where $\mathbf{z} = (\mathbf{g} - \mathbf{b})$. The partial derivatives will be expressed in equations (F.22) and (F.23).

The Hessian matrix is given by [202]:

$$\mathbf{G}_k \triangleq \nabla^2_{a(i)}\xi_k^{RLS} = \frac{\partial^2 \xi_k^{RLS}}{\partial a^2(i)} = \lambda\hat{\mathbf{G}}_{k-1} + \frac{\partial^2 \varepsilon_k}{\partial a^2(i)}. \tag{F.19}$$

The second order gradient of the error needed above is obtained as:

$$\frac{\partial^2 \varepsilon}{\partial a^2(i)} = \frac{2}{N}\left\{\left[\frac{\partial^2 \boldsymbol{\epsilon}}{\partial a^2(i)}\right]^{\mathrm{T}} \boldsymbol{\epsilon} + \frac{\partial \boldsymbol{\epsilon}}{\partial a(i)}\left[\frac{\partial \boldsymbol{\epsilon}}{\partial a(i)}\right]^{\mathrm{T}}\right\}. \tag{F.20}$$

Applying the second order gradient to the estimation error, one obtains:

$$\frac{\partial^2 \boldsymbol{\epsilon}}{\partial a^2(i)} = -\mathbf{z}^{\mathrm{T}}\left[2\frac{\partial \mathbf{A}}{\partial a(i)}\mathbf{M}\frac{\partial \mathbf{A}^{-1}}{\partial a(i)} + \mathbf{A}\mathbf{M}\frac{\partial^2 \mathbf{A}^{-1}}{\partial a^2(i)}\right]^{\mathrm{T}}. \tag{F.21}$$

In order to aid the visualization of the equations, we will use $\bar{\mathbf{A}} = \mathbf{A}^{-1}$. Moreover, the higher order derivatives of the gain matrix $\mathbf{A}$ and its inverse are presented as:

$$\dot{\mathbf{A}}(i) = \frac{\partial \mathbf{A}}{\partial a(i)} = \begin{bmatrix} \mathbf{0} & & \\ & 1 & \\ & & \mathbf{0} \end{bmatrix} \tag{F.22}$$

$$\dot{\bar{\mathbf{A}}}(i) = \frac{\partial \mathbf{A}^{-1}}{\partial a(i)} = \begin{bmatrix} \mathbf{0} & & \\ & -a^{-2}(i) & \\ & & \mathbf{0} \end{bmatrix} \tag{F.23}$$

and

$$\ddot{\bar{\mathbf{A}}}(i) = \frac{\partial^2 \mathbf{A}^{-1}}{\partial a^2(i)} = \begin{bmatrix} \mathbf{0} & & \\ & 2a^{-3}(i) & \\ & & \mathbf{0} \end{bmatrix}, \tag{F.24}$$

where only the ii-th elements differ from zero. Then, equation (F.18) can be written more compactly as

$$\nabla_{a(i)}\varepsilon = -\mathbf{z}^{\mathrm{T}}\left[\dot{\mathbf{A}}(i)\mathbf{M}\bar{\mathbf{A}} + \mathbf{A}\mathbf{M}\dot{\bar{\mathbf{A}}}(i)\right]^{\mathrm{T}}\boldsymbol{\epsilon}. \tag{F.25}$$

The Tensorial-RLS algorithm for gain estimation is shown in Table F.2. The "tensorial" denomination comes from the fact that the solution for whole image needs tensorial notation, as the solution for individual pixels uses matrix-vector and matrix-matrix multiplications.

Vectorization of Tensorial-RLS Algorithm for Gain Correction

Due to sparse structure of matrices in the Tensorial-RLS algorithm, it is possible to group the calculations in order to transform the loop (in the Tensorial-RLS algorithm of table F.2) into matrix operations. This can lead to a significant improvement in the speed of the algorithms when implemented in a matrix-oriented programming language such as MATLAB® .

Do for $k \geq 0$

$\mathbf{z}_k = \left(\mathbf{g}_{k-1} - \mathbf{b}\right)$

$\boldsymbol{\epsilon}_k = \mathbf{g}_k - \mathbf{A}_k\mathbf{M}_k\bar{\mathbf{A}}_k\mathbf{z}_k - \mathbf{b}$

Do for $1 \leq i \leq N$

$\mathbf{u}_k(i) = \left[\dot{\mathbf{A}}_k(i)\mathbf{M}_k\bar{\mathbf{A}}_k + \mathbf{A}_k\mathbf{M}_k\dot{\bar{\mathbf{A}}}_k(i)\right]\mathbf{z}_k$

$\mathbf{w}_k(i) = \left[2\dot{\mathbf{A}}_k(i)\mathbf{M}_k\dot{\bar{\mathbf{A}}}_k(i) + \mathbf{A}_k\mathbf{M}_k\ddot{\bar{\mathbf{A}}}_k(i)\right]\mathbf{z}_k$

$v_k(i) = \mathbf{w}_k^{\mathrm{T}}(i)\boldsymbol{\epsilon}_k + \mathbf{u}_k^{\mathrm{T}}(i)\mathbf{u}_k(i)$

$g_k(i) = \lambda g_{k-1}(i) + v_k(i)$

$\hat{a}_{k+1}(i) = \hat{a}_k(i) - g_k^{-1}(i)\mathbf{u}_k^{\mathrm{T}}(i)\boldsymbol{\epsilon}_k$

Table F.2: Tensorial-RLS algorithm for gain correction

First, we define a matrix $\mathbf{Z} = \mathrm{diag}\,(\mathbf{z})$ and we redefine (k index hidden to aid visualization)

$$\dot{\bar{\mathbf{A}}} = -\mathrm{diag}\left(a^{-2}(1), a^{-2}(2), \cdots, a^{-2}(N)\right) \tag{F.26}$$

$$\ddot{\bar{\mathbf{A}}} = 2\mathrm{diag}\left(a^{-3}(1), a^{-3}(2), \cdots, a^{-3}(N)\right), \tag{F.27}$$

which are calculated at each iteration.

Let's first analyze the term $\mathbf{u}_k(i)$ in table F.2. Its squared norm will be used to calculate $v_k(i)$. However, we can compute $\mathbf{u}_k(i)$ for all i in one step, store the results in the columns of a matrix $\mathbf{U}_k$ and then compute their squared norms. The latter operation will be represented by $\sum_i^N \left[\mathbf{U}_k \circ \mathbf{U}_k\right]_{ij}$, meaning that the norms of the columns are calculated and stored in a row-vector. The symbol '$\circ$' represents Hadamard or element-wise product, that is, $\mathbf{p} = \mathbf{q} \circ \mathbf{r}$ is given by $[\mathbf{p}]_i = [\mathbf{q}]_i\,[\mathbf{r}]_i$.

The same idea can be applied to the term $\mathbf{w}_k(i)$ in table F.2. The results for all i will be calculated in one step and stored in the matrix $\mathbf{W}_k$, which will be further multiplied by $\boldsymbol{\epsilon}_k$. It is important to note that the conception of matrices $\mathbf{U}_k$ and $\mathbf{W}_k$ is possible owing to the special sparsity of the matrices $\dot{\mathbf{A}}_k(i)$, $\dot{\bar{\mathbf{A}}}_k(i)$ and $\ddot{\bar{\mathbf{A}}}_k(i)$. They have only the ii-th element different from zero, thus a post-multiplication by a matrix (e.g. $\dot{\mathbf{A}}_k(i)\mathbf{M}_k\bar{\mathbf{A}}_k$) will conserve only the i-th row, whereas a pre-multiplication (e.g. $\mathbf{A}_k\mathbf{M}_k\dot{\bar{\mathbf{A}}}_k(i)$) will keep only the i-th column.

The vector version of Tensorial-RLS algorithm for gain correction is shown in Table F.3, where Hadamard or element-wise division $\mathbf{p} = \left(\frac{\mathbf{q}}{\mathbf{r}}\right)$ is given by $[\mathbf{p}]_i = \frac{[\mathbf{q}]_i}{[\mathbf{r}]_i}$.

Do for $k \geq 0$

$$\mathbf{z}_k = \left(\mathbf{g}_{k-1} - \mathbf{b}\right)$$
$$\boldsymbol{\epsilon}_k = \mathbf{g}_k - \mathbf{A}_k\mathbf{M}_k\bar{\mathbf{A}}_k\mathbf{z}_k - \mathbf{b}$$
$$\mathbf{Z}_k = \mathrm{diag}\left(\mathbf{z}_k\right)$$
$$\mathbf{U}_k = \mathrm{diag}\left(\mathbf{M}_k\bar{\mathbf{A}}_k\mathbf{z}_k\right) - \mathbf{Z}_k\mathbf{A}_k\mathbf{M}_k\dot{\bar{\mathbf{A}}}_k$$
$$\mathbf{W}_k = \mathbf{Z}_k\left(2\mathbf{M}_k\dot{\bar{\mathbf{A}}}_k - \mathbf{A}_k\mathbf{M}_k\ddot{\bar{\mathbf{A}}}_k\right)$$
$$\mathbf{v}_k = \mathbf{W}_k\boldsymbol{\epsilon}_k + \left[\sum_i \left[\mathbf{U}_k \circ \mathbf{U}_k\right]_{ij}\right]^{\mathrm{T}}$$
$$\mathbf{g}_k = \lambda\mathbf{g}_{k-1} + \mathbf{v}_k$$
$$\boldsymbol{\alpha}_k = \frac{\mathbf{U}_k^{\mathrm{T}}\boldsymbol{\epsilon}_k}{\mathbf{g}_k}$$
$$\mathbf{A}_k = \mathbf{A}_{k-1} + \mathrm{diag}\left(\boldsymbol{\alpha}_k\right)$$

Table F.3: Tensorial-RLS algorithm for gain correction (vector form)

F.6 Affine Projection algorithms

Affine projection (AP) is a class of adaptive-filtering algorithms which recycles the old data signal in order to improve the convergence as compared to stochastic gradient-type of algorithms. Also referred to as data-reusing algorithms, the AP algorithms are known to be viable alternatives to the RLS algorithms by achieving lower computational complexity in situations where the input signal is correlated. The penalty to be paid when increasing the number of data reuse is a slight increase in algorithm misadjustment [203].

Due to memory limitations in the implementation of AP algorithm, we introduce here a different approach from the one usually found in the literature [199, 204, 205]. We define the objective function as

$$\xi_k^{AP} = \sum_{i=k-L}^{k} \varepsilon_i = \sum_{i=k-L}^{k} \boldsymbol{\epsilon}_i^{\mathrm{T}}\boldsymbol{\epsilon}_i, \tag{F.28}$$

where L corresponds to the amount of reused data. By minimizing (F.28) we minimize the estimation error squared over a window of size L. The main difference between RLS and AP algorithm is that the former considers the whole past of errors weighted by the forgetting factor, whereas the latter considers only a window of past errors, giving the same weight to all errors.

The AP algorithm usually requires less computational complexity than the RLS algorithm brought about by the reduction in the dimension of the information matrix that is inverted. In addition, the finite memory of the AP algorithm reduces the noise enhancement and the negative effects of the slow variations of the FPN, both inherent to the RLS algorithm, see [199] for details.

Do for $k \geq 0$

$\boldsymbol{\epsilon}_k = \mathbf{g}_k - \mathbf{A}\mathbf{M}_k\mathbf{A}^{-1}\left(\mathbf{g}_{k-1} - \hat{\mathbf{b}}_k\right) - \hat{\mathbf{b}}_k$

$\mathbf{R}_k = \left[\left(\mathbf{A}\mathbf{M}_k\mathbf{A}^{-1}\right)^{\mathrm{T}} - \mathbf{I}\right] \cdot \left[\left(\mathbf{A}\mathbf{M}_k\mathbf{A}^{-1}\right) - \mathbf{I}\right]$

Keep $\mathbf{R}_k, \mathbf{R}_{k-1}, \cdots, \mathbf{R}_{k-L-1}$ in memory.

$\hat{\mathbf{H}}'_k = \lambda\hat{\mathbf{H}}'_{k-1} + \mathbf{R}_k - \mathbf{R}_{k-L-1}$

$\mathbf{u}_k = \left[\left(\mathbf{A}\mathbf{M}_k\mathbf{A}^{-1}\right)^{\mathrm{T}} - \mathbf{I}\right] \boldsymbol{\epsilon}_k$

$\mathbf{v}_k = \mathrm{BCGSTABL}(\hat{\mathbf{H}}'_k, \mathbf{u}_k)$

$\hat{\mathbf{b}}_{k+1} = \hat{\mathbf{b}}_k - \mathbf{v}_k$

Table F.4: AP algorithm for bias correction

Following a similar procedure to the one used in Section F.5, it can be shown that the Hessian matrix can be estimated by

$$\begin{aligned} \mathbf{H}_k \triangleq \nabla^2_{\mathbf{b}}\xi_k^{AP} &= \frac{\partial^2 \xi_k^{AP}}{\partial \mathbf{b} \partial \mathbf{b}^{\mathrm{T}}} \\ &= \hat{\mathbf{H}}_{k-1} + \frac{\partial^2 \varepsilon_k}{\partial \mathbf{b} \partial \mathbf{b}^{\mathrm{T}}} - \frac{\partial^2 \varepsilon_{k-L-1}}{\partial \mathbf{b} \partial \mathbf{b}^{\mathrm{T}}}. \end{aligned} \tag{F.29}$$

The matrix $\hat{\mathbf{H}}_k$ accumulates information about the last L errors. When new information comes, the oldest error contribution has to be subtracted. In short, RLS and AP algorithms differ from the way the Hessian matrix is estimated. Apart from that, the algorithms are basically the same (e.g. *a priori* error gradient, etc).

F.6.1 Bias correction by AP algorithm

The update equation for the Affine Projection algorithm is the same as equation (F.9), repeated here for convenience:

$$\hat{\mathbf{b}}_{k+1} = \hat{\mathbf{b}}_k - \hat{\mathbf{H}}_k^{-1} \nabla_{\mathbf{b}} \varepsilon_k, \tag{F.30}$$

where $\hat{\mathbf{H}}_k$ is an estimate of the Hessian matrix and $\nabla_{\mathbf{b}}\varepsilon_k$ is the *a priori* error gradient[199].

It is straightforward to show, by substituting equations (F.12) and (F.13) into equation (F.29), that

$$\hat{\mathbf{H}}'_k = \lambda\hat{\mathbf{H}}'_{k-1} + \mathbf{R}_k - \mathbf{R}_{k-L-1}, \tag{F.31}$$

where $\mathbf{R}_k = \left[\left(\mathbf{A}\mathbf{M}_k\mathbf{A}^{-1}\right)^{\mathrm{T}} - \mathbf{I}\right] \cdot \left[\left(\mathbf{A}\mathbf{M}_k\mathbf{A}^{-1}\right) - \mathbf{I}\right]$. Past values of $\mathbf{R}_k$, up to the $(k-L-1)$-th, must be kept in memory.

The complete affine projection bias correction algorithm is summarized in Table F.4.

Do for $k \geq 0$

$\mathbf{z}_k = \left(\mathbf{g}_{k-1} - \mathbf{b}\right)$

$\boldsymbol{\epsilon}_k = \mathbf{g}_k - \mathbf{A}_k\mathbf{M}_k\bar{\mathbf{A}}_k\mathbf{z}_k - \mathbf{b}$

$\mathbf{U}_k = \mathrm{diag}\left(\mathbf{M}_k\mathbf{A}_k\mathbf{z}_k\right) - \mathbf{Z}_k\mathbf{A}_k\mathbf{M}_k\dot{\bar{\mathbf{A}}}_k$

$\mathbf{W}_k = \mathbf{Z}_k\left(2\mathbf{M}_k\dot{\bar{\mathbf{A}}}_k - \mathbf{A}_k\mathbf{M}_k\ddot{\bar{\mathbf{A}}}_k\right)$

$\mathbf{v}_k = \mathbf{W}_k\boldsymbol{\epsilon}_k + \left[\sum_i \left[\mathbf{U}_k \circ \mathbf{U}_k\right]_{ij}\right]^{\mathrm{T}}$

Keep $\mathbf{v}_k, \mathbf{v}_{k-1}, \cdots, \mathbf{v}_{k-L-1}$ in memory.

$\mathbf{g}_k = \lambda\mathbf{g}_{k-1} + \mathbf{v}_k - \mathbf{v}_{k-L-1}$

$\boldsymbol{\alpha}_k = \dfrac{\mathbf{U}_k^{\mathrm{T}}\boldsymbol{\epsilon}_k}{\mathbf{g}_k}$

$\mathbf{A}_k = \mathbf{A}_{k-1} + \mathrm{diag}\left(\boldsymbol{\alpha}_k\right)$

Table F.5: Tensorial-AP algorithm for gain correction (vector form)

F.6.2 Gain correction by Tensorial-AP algorithm

For the gain estimation, the vector $\mathbf{g}_k$ in Table F.3 plays the role of Hessian matrix. Strictly speaking, as seen in Section F.5.3, the second-order partial derivatives become scalars and we have only to worry about one value per pixel. The values of $\mathbf{g}_k$ can be regarded as variable step sizes for each pixel. Their update rule follows the same idea of Section F.6.1: the newest $\mathbf{v}_k$ of L values will be added to the accumulator $\mathbf{g}_k$, whereas the oldest (i.e. $\mathbf{v}_{k-L-1}$) will be subtracted from the accumulator.

In the experiments described in Section F.7, we use the vector form of the Tensorial-RLS algorithm (Table F.3) as a basis to Tensorial-AP algorithm. The difference is in how the vector $\mathbf{g}_k$ is updated. The complete Tensorial-AP algorithm for gain correction is shown in Table F.5.

F.6.3 Handling dead pixels, leaking pixels and algorithm breakdown

In this section we address problematic conditions which can lead to an algorithmic breakdown. We base our analysis primarily on equations (F.1) and (F.2). A breakdown would occur if we could not use equation (F.2) for FPN correction, repeated here for convenience:

$$\hat{f}_k(n_1, n_2) = \frac{g_k(n_1, n_2) - \hat{b}(n_1, n_2)}{\hat{a}(n_1, n_2)}. \tag{F.32}$$

Suppose we apply our FPN correction method to an IR video sequence obtained by a camera with dead pixels, i.e. $a(n_1, n_2) = 0$ for some $\{i, j\}$. The gain estimation algorithm would eventually converge to $\hat{a}(n_1, n_2) = 0$ and equation (F.32) could not

be used. Thus, we could set a minimum accepted gain, say a_{min}, and monitor all $\hat{a}(n_1, n_2)$ to ensure that their values are at least a_{min}.

Another situation where the video acquisition does not agree with the observation model in (F.1), is the case of leaking pixels, i.e. $g_k(n_1, n_2) = a(n_1, n_2)f(n_1, n_2) + b(n_1, n_2) + a(i + \delta_i, j + \delta_j)f(i + \delta_i, j + \delta_j) + b(i + \delta_i, j + \delta_j)$ for some $\{i, j, \delta_i, \delta_j\}$. In this case, divergence of $\hat{a}(n_1, n_2)$ and $\hat{b}(n_1, n_2)$ estimates may occur.

Ideally, in order to take into account these particularities, the camera sensor should be studied and its model used to derive the algorithms. However, we can prevent breakdowns by adequately choosing $\{a_{min}, a_{max}\}$ and $\{b_{min}, b_{max}\}$ and simply forcing $\hat{a}(n_1, n_2)$ and $\hat{b}(n_1, n_2)$ to lie within these ranges.

F.7 Results

This section presents the results obtained with the proposed algorithms and compares their performances to the state-of-the-art NUC algorithms. First, the performance – meaning fidelity of estimated video to the uncorrupted video – of the algorithms is assessed through simulated data. Synthetic FPN and random noise are introduced in simulated infrared video obtained from a static image. Then, we apply the algorithms to real FPN-corrupted infrared video, where the performance of the methods is subjectively evaluated. In the experiments pixel values can only range from 0 to 255 (image dynamic range).

F.7.1 Simulation Results

For image quality evaluation, we use two measures: PSNR[1], for its frequent use in image quality assessment, and SSIM (Structural SIMilarity) [206], for its good consistency with subjective assessment compared to other measures. Both PSNR and SSIM indicate how close the estimated image $\hat{\mathbf{f}}_k$ is from the real uncorrupted image $\mathbf{f}_k$. When the two images are identical the PSNR will tend to infinite, whereas SSIM will be one. Thus, the higher both measures are, the better is the image's fidelity. We will use $\log_{10}(\text{SSIM})$ to emphasize the numerical differences between the methods.

We compared four algorithms: LMS-based NUC developed in [4], Kalman-Filter-based [192], Tensorial-RLS [97] (described in detail in Section F.5) and the proposed Tensorial-AP described in Section F.6. We have chosen the Kalman-Filter-based method described in [192] as the reference method for our comparisons since it provides state-of-the-art results without any assumptions about the scene content or motion behavior. Other methods may provide similar results, but they often rely on

[1] $\text{PSNR}(\mathbf{x}, \mathbf{y}) = 10 \log_{10} \frac{255^2}{\text{MSE}(\mathbf{x}, \mathbf{y})}$

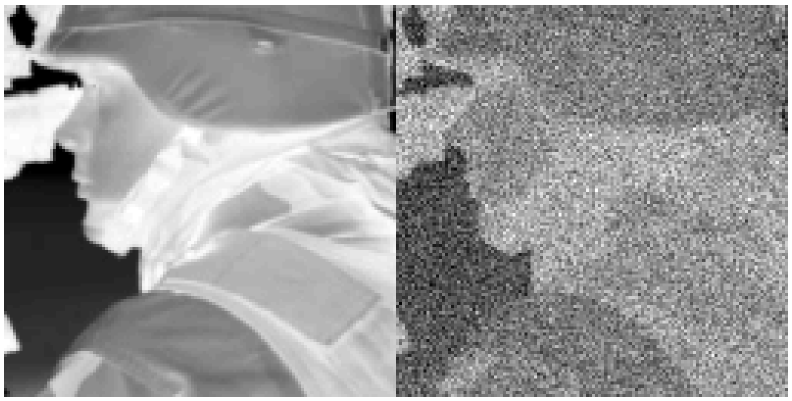


Figure F.2: Original image (left) and image corrupted with synthetic FPN – fixed-pattern noise (right)

motion constraints (i.e. only 1D in [91] or non-subpixel in [191]), which restricts the gamut of videos they can be applied to. Also, Zuo et al. [207] proposed a gradient descent NUC algorithm which considers motion explicitly. The method is similar to the LMS-based one developed in [4].

We generated 50 videos from portions of static images. An example can be seen in Figure F.2. Each video contained 250 frames with resolution 128×128 pixels. Between consecutive frames there were translational shifts defined by random real numbers from -2 to 2. Synthetic FPN was inserted and corrupted all frames of the videos. We remind that FPN (fixed-pattern noise), as the name suggests, is time-invariant. We inserted FPN according to equation (F.1) with gain standard deviation randomly selected from the interval $0 \leq \sigma_{\mathbf{A}} \leq 0.1$ and bias standard deviation randomly selected from the interval $0 \leq \sigma_{\mathbf{b}} \leq 0.5$. Additive noise, with standard deviation randomly selected from the interval $0 \leq \sigma_{\mathbf{n}} \leq 0.05$, was also added to each frame. We used normal distribution for the random selection of the bias and the gain.

We have used $\mu = 0.1$ as step size in the LMS-based algorithm, $\lambda = 0.999$ as forgetting factor in the RLS algorithm and $L = 3$ as the number of reused inputs in the AP algorithm.

The videos were processed with the four algorithms, and the fidelity of the reconstructed video in reference to the uncorrupted one was evaluated through PSNR and SSIM measures. Figures F.3 and F.4 show the average of the 50 generated videos. The frame number axis was not averaged in order to show the convergence of all methods. Table F.6 shows the average results.

Additionally, the experiments have shown that the order that gain and bias FPN correction are performed (i.e. first bias then gain or first gain then bias) does not affect the final result. As at each iteration (i.e. each new frame) the bias and gain estimates are only slightly refined, the final result does not depend on the order of

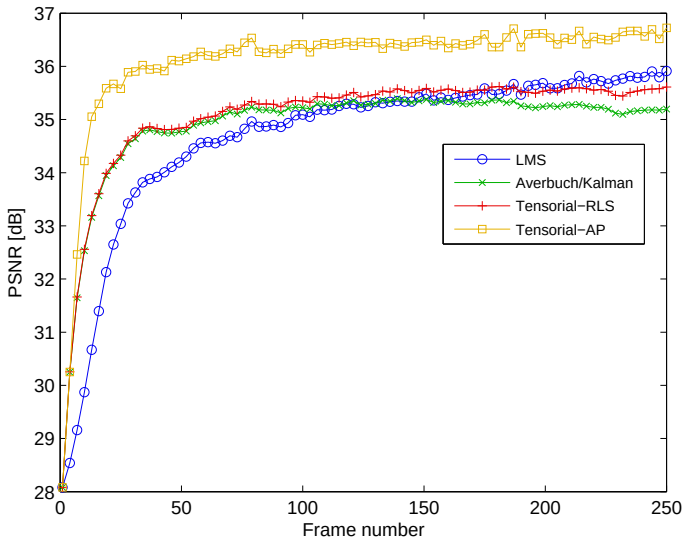


Figure F.3: Mean PSNR results from simulated data

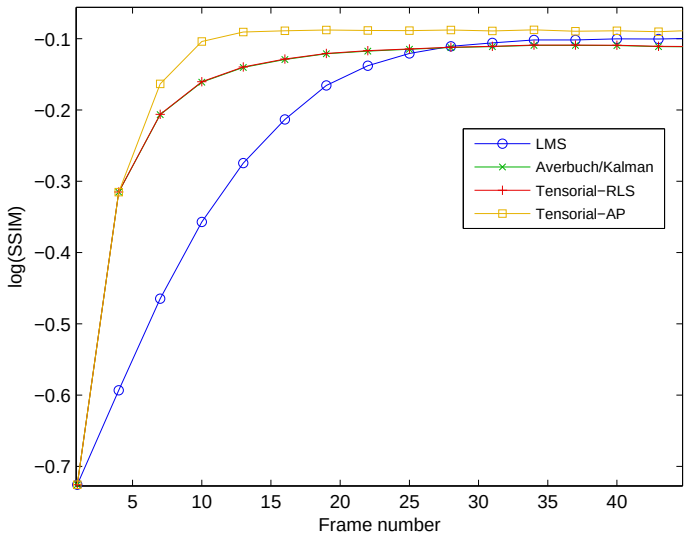


Figure F.4: Mean SSIM results from simulated data

	PSNR [dB]	log(SSIM)[$\times 10^{-4}$]
LMS	34.7043	-0.1226
Kalman	34.9060	-0.1242
Tensorial-RLS	35.0844	-0.1234
Tensorial-AP	36.1221	-0.0989

Table F.6: Average results for 50 synthetically FPN-corrupted videos. Higher values of PSNR and SSIM denote better results.

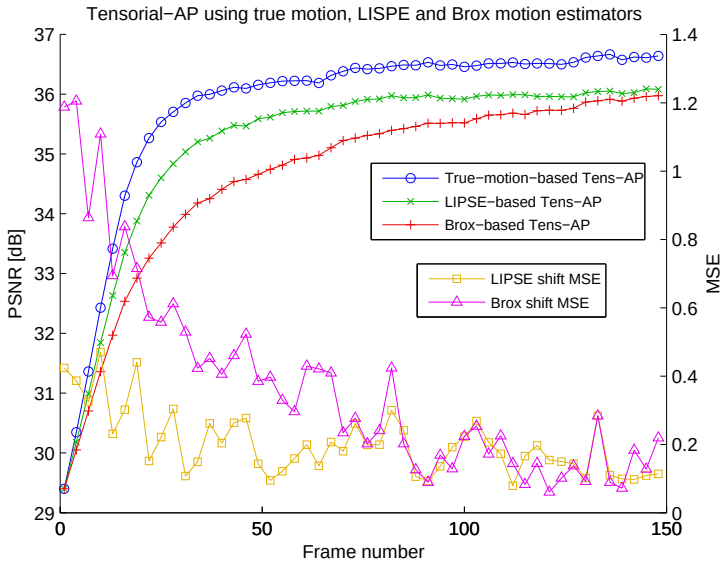


Figure F.5: Tensorial-AP performance using true motion information, shift estimation by LIPSE algorithm and by Brox [208] algorithm

corrections.

F.7.2 Errors in Shift Estimation

In this section, we assess the performance of the Tensorial-AP algorithm in terms of shift estimation errors and shift estimation algorithms. We generated 100 videos from portions of static images with known vertical and horizontal shifts. Then, we fed the Tensorial-AP algorithm with true motion information, motion estimated by LIPSE algorithm described in Section F.8 and motion estimated by Brox's algorithm described in [208]. Since the FPN estimation and removal improves according to the frame number, we applied a pre-correction before estimating shifts between each pair of frames. Therefore, shift estimation error also tends to improve with time for the noise level tends to decrease. Figure F.5 shows this behavior and the evolution of image quality and motion estimation mean squared error (MSE) with time.

As expected, the best performance is attained when true motion is available. However, LIPSE algorithm performed better than Brox's [208], which is one of a

LMS	Step size	$\mu = 0.3$
Kalman	Forgetting factor	$\lambda = 0.9$
Tensorial-RLS	Forgetting factor	$\lambda = 0.9$
Tensorial-AP	Reuse order & step size	$L = 2$ & $\mu = 2$

Table F.7: Parameters used for each algorithm in the experiments with real infrared videos.

family of optical-flow-based algorithms [118, 208–211].

Although optical-flow-based algorithms provide excellent results for most real videos, when strong fixed-pattern noise is present the averaging nature of LIPSE (see Section F.8) attenuates the FPN and provides better results, mainly in the beginning of the simulations, when the FPN level is high, as argued in [91, 212]. Moreover, in our studies we have focused on pure translational motion, justifying the use of LIPSE algorithm. For complex non-global motion, a differential coarse-to-fine motion estimation method should be used [118, 208–211].

Figure F.5 also shows the performance of Tensorial-AP algorithm in presence of errors in shift estimation. The convergence speed is slowed down when motion is not accurately estimated. However, since we use only one iteration of the recursive algorithm for each new incoming frame, each update is only slightly affected. Moreover, when the noise level lowers with time, the shift estimation tends to become more accurate.

F.7.3 Real IR Videos Results

We shot video sequences using a FLIR SYSTEMS model ThermaCAM P65 infrared camera, with a *focal plane array uncooled microbolometer* detector. Each infrared sequence consists of 200 frames with picture size of 320×240 pixels at 60 frames/second. The "*Noise Reduction*" option was switched off, as well as the "*Shutter Period*" option. The latter refers to the FPN correction provided by the camera manufacturer. When active, the camera shutter receives a periodic signal closure (varying from 3 to 15 minutes) to perform a two-point calibration.

The FPN contamination was very clear in the acquired video. In order to assess the performance of each FPN reduction method, we applied the four algorithms under evaluation to the captured videos. The algorithms had some of their parameters empirically adjusted in order to achieve the best result from each method. The new parameters are shown in Table F.7.

Figures F.6 to F.9 show the 133-th frame of the observed video "telephone" alongside the corrected videos by each of the four algorithms under evaluation. Figures F.10 to F.13 show the 190-th frame of the observed video "tube segment"

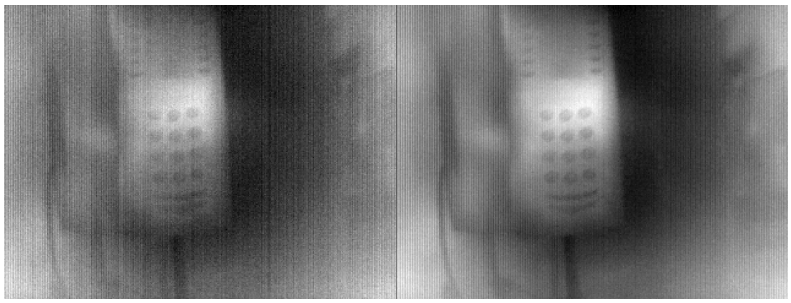


Figure F.6: Telephone observed IR video (left) and FPN corrected video by LMS algorithm

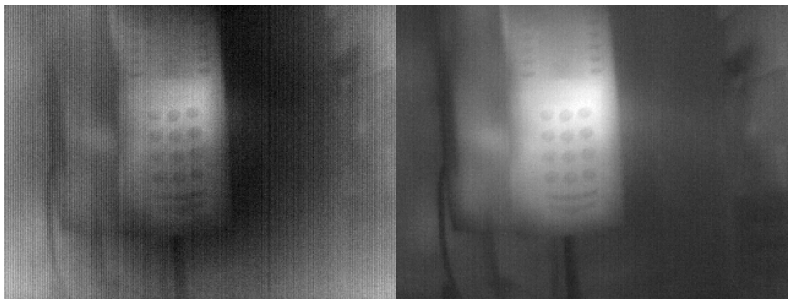


Figure F.7: Telephone observed IR video (left) and FPN corrected video by Averbuch/Kalman algorithm

alongside the corrected videos by each of the four algorithms under evaluation.

We also applied the Tensorial-AP algorithm to a infrared video sequence obtained on the internet (`http://www.youtube.com/watch?v=lHw_JWLkqOo`). The sequence shows aerial images of a truck and another vehicle. The images are corrupted with FPN, though one can notice that it is a different type of FPN. Specifically, the stripe pattern is horizontal rather vertical. The 69-second original frame (on the left) along with the processed image (on the right) are shown in Figure F.14 as an example of the Tensorial-AP algorithm output.

We can observe on the real images that the algorithms were able to remove the FPN more or less efficiently depending on each one's characteristics. Interestingly, the video on Figure F.14 showed originally overlaid helping information and target lines which were also removed by the Tensorial-AP algorithm. Considering that those artifacts are constant throughout the entire sequence, they match the FPN definition and their removal is consistent.

From a mathematical perspective, the results obtained are reasonable since we do not assume in our models any vertical or horizontal pattern for the FPN. Rather, each pixel has its own gain and bias and no correlation between pixels is imposed.

As can be observed, the best perceptual results are achieved by the AP and RLS algorithms with the former having a much smaller computational complexity, as detailed in the next subsection.

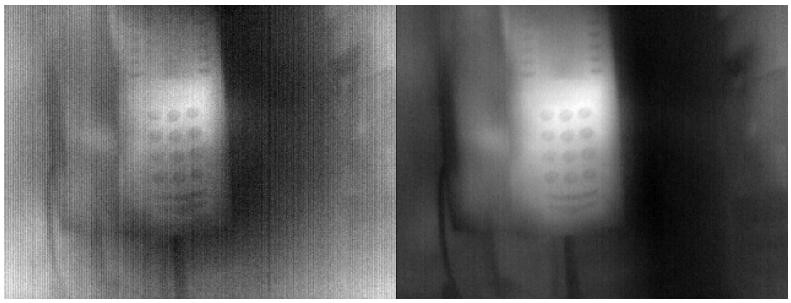


Figure F.8: Telephone observed IR video (left) and FPN corrected video by Tensorial-RLS algorithm

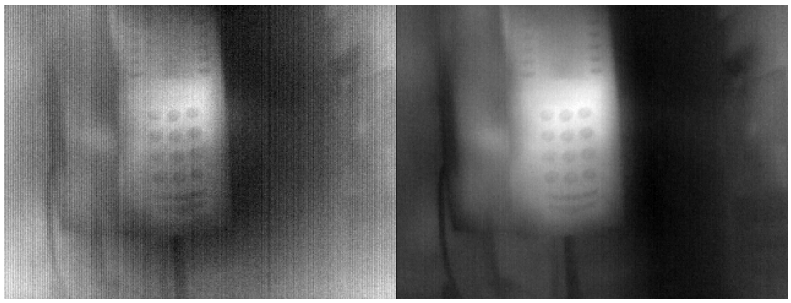


Figure F.9: Telephone observed IR video (left) and FPN corrected video by Tensorial-AP algorithm

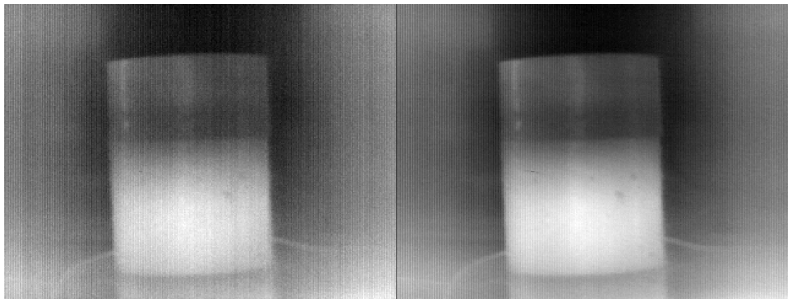


Figure F.10: Tube segment observed IR video (left) and FPN corrected video by LMS algorithm

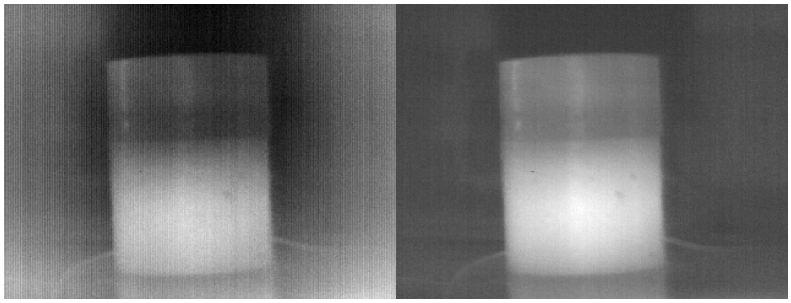


Figure F.11: Tube segment observed IR video (left) and FPN corrected video by Averbuch/Kalman algorithm

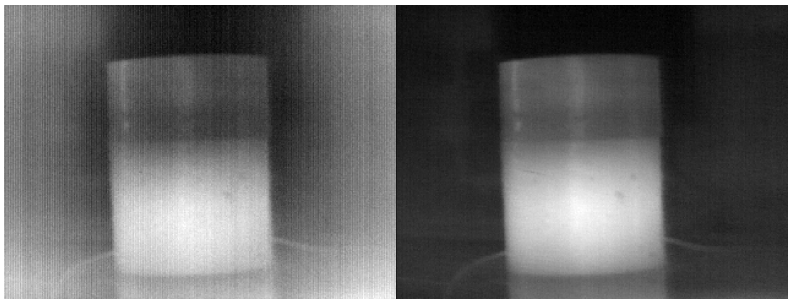


Figure F.12: Tube segment observed IR video (left) and FPN corrected video by Tensorial-RLS algorithm

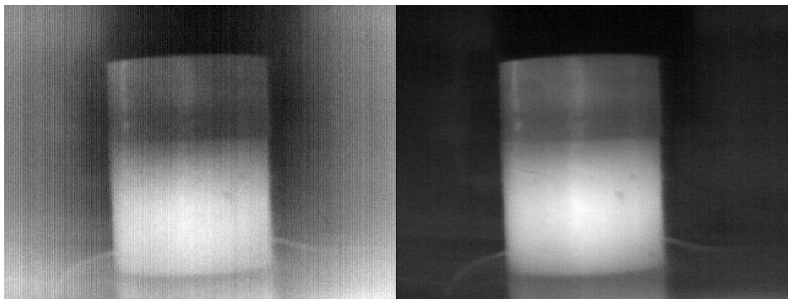


Figure F.13: Tube segment observed IR video (left) and FPN corrected video by Tensorial-AP algorithm

Figure F.14: Observed "truck" IR video (left) and FPN corrected video by Tensorial-AP algorithm

LMS	Kalman	Tensorial-RLS	Tensorial-AP
26.7441	315.7993	334.2236	76.3300

Table F.8: Average time (in seconds) of execution for 50 synthetically FPN-corrupted videos. Lower values are better.

F.7.4 Computational load

The computational complexity of each algorithm has also been assessed by measuring their execution times. Table F.8 shows the average results in seconds for the 50 videos mentioned in Section F.7.1. As expected, LMS-based algorithm was the fastest due to its simplicity. Tensorial-AP outperformed the others because BCGSTABL converged in less steps.

F.8 LIPSE motion estimation algorithm

LIPSE stands for "linear interpolation projection-based shift estimator". The advantages of projection-based algorithms are speed and noise robustness, specially the FPN which can severely affect motion estimation reliability [212]. For detail information on LIPSE, see [91, 212].

Let $\mathbf{g}_{k-1}$ and $\mathbf{g}_k$ be two consecutive frames of a video presenting only translation shifts. Each pixel of the image is represented by $g_T(n_1, n_2)$, and the projections of rows and columns are respectively defined as $g_R(n_2) = \frac{1}{N_1}\sum_{n_1=1}^{N_1} g_T(n_1, n_2)$ and $g_C(n_1) = \frac{1}{N_2}\sum_{n_2=1}^{N_2} g_T(n_1, n_2)$.

The solution will be given only to the projection of columns ($y_C(n_1)$, referred to as $y(n)$ from now on), since it is analogous to the projection of rows. Suppose that there is only subpixel motion between consecutive frames (as integer shifts are compensated – see algorithm below). Each element of the projection of the k-th frame is estimated by

$$\hat{g}_k(n) = (1 - \delta_k)g_{k-1}(n) + \delta_k g_{k-1}(n+1), \tag{F.33}$$

where $0 \leq \delta_k < 1$ is the subpixel shift. The MSE (mean-square error) is defined as $\varphi_k = \frac{1}{N}\sum_{n=1}^{N}\left[g_k(n) - \hat{g}_k(n)\right]^2$.

The value of δ_k which minimizes the MSE is such that $\frac{\partial \varphi_k}{\partial \delta_k} = 0$ and the solution is given by $\delta_k = \frac{\psi_k}{\zeta_k}$, where

$$\psi_k = \sum_{n=1}^{N-1}\Big\{g_k(n)\left[g_{k-1}(n) - g_{k-1}(n+1)\right] + g_{k-1}(n)\left[g_{k-1}(n+1) - g_{k-1}(n)\right]\Big\} \tag{F.34}$$

and

$$\zeta_k = \sum_{n=1}^{N-1}\Big\{g_{k-1}^2(n) - 2g_{k-1}(n)g_{k-1}(n+1) + g_{k-1}^2(n+1)\Big\}. \tag{F.35}$$

In summary, the LIPSE algorithm is given by the following steps:

1. Compute $\delta_k = \frac{\psi_k}{\zeta_k}$ through equations (F.34) and (F.35) for all possible integer shifts Δ_k between the projections of consecutive frames $y_R(j)$ and $y_C(j)$;
2. Find MSE's φ_k for all combination produced in the previous step;
3. Select the Δ_k which produces the smallest MSE φ_k;
4. Form the total shift estimate $d_k = \Delta_k + \delta_k$;
5. Repeat previous steps for the projections of the rows $y_R(j)$ and then obtain the complete shifts (columns and rows) $\mathbf{d}_k$.

F.9 Conclusion

This chapter presented two new algorithms for NUC (bias and gain nonuniformity correction) in infrared videos. The proposed methods, called Tensorial-RLS and Tensorial-AP, are based on Recursive Least-Squares and Affine Projection adaptive filters. They received the "Tensorial" denomination due to the fact that their derivation includes the concept of tensors.

Although the notion of tensors was employed, it has not been necessary to use tensorial notation. Instead, a pixel-by-pixel version was developed, which was further grouped into a compact vectorial version. This version is easier to implement and faster when matrix-oriented programs are used (e.g. in MATLAB, matrix multiplications are faster than loops).

Section F.7 showed the results when comparing the proposed algorithms to state-of-the-art NUC proposed by Averbuch in [192]. Although Averbuch named his algorithm "Kalman-filter-based", it is rather an RLS-based method, as Averbuch himself observed. In fact, when only bias FPN is present, the performances of Tensorial-RLS and Averbuch/Kalman algorithm are quite equivalent.

However, when also gain FPN exists, Tensorial-RLS outperforms Kalman-based methods, as the former corrects both gain and bias FPN whereas the latter corrects bias only. The affine projection algorithm (Tensorial-AP), on the other hand, showed the best results of video quality for both image quality measures (PSNR and SSIM).

Concerning speed of convergence and final misadjustment, the experiments showed the following:

- LMS (only bias): slow convergence and low misadjustment;
- Kalman (only bias): fast convergence and high misadjustment;
- RLS (bias and gain): fast convergence and high misadjustment;
- AP (bias and gain): fast convergence and low misadjustment.

Both RLS and Kalman presented high misadjustment probably due to the additive random noise incorporated (i.e. convergence and final misadjustment trade-off). Affine projection algorithm showed good speed of convergence and low misadjustment compared with RLS and had the best combined results of the compared methods.

Observing the results from the experiments using real infrared video, we can subjectively rank each method according to image quality as (informal subjective tests have been carried out): Tensorial-RLS and Tensorial-AP (best results), Kalman and LMS-based (worst result). The Tensorial-RLS and the Tensorial-AP algorithms were considered equivalent in terms of subjective quality.

■[2]

[2]Conduz-se o leitor de volta ao corpo da tese no Capítulo 8, página 42.

Appendix G

Survey of multi-frame super-resolution methods

The aim of multi-frame super-resolution is to recover HR images from a set of LR observed images. The LR observed images can be, for instance, several pictures of an object or frames of a video. In this work, we will consider the latter case.

In most real videos, it is possible to find mathematical relations between frames. Many video coding schemes adopt relative motion models between frames in order to describe local motion of objects in the scene or global motion of the whole scene. In the simplest case, the differences between the LR observations are due to translational motion of the camera. The motion parameters, that is, the amount of vertical and horizontal displacement of camera, are assumed to be known in most cases. Although these parameters can also be obtained by motion estimation algorithms, this is a serious drawback since the motion parameters in HR have to be estimated from the LR observed frames. Specifically, the motion estimation must have sub-pixel accuracy.

Multi-frame super-resolution methods aim to enhance the resolution of a reference frame by exploiting relationships between frames. Other frames can contain additional information about the reference frame. In this case, the additional information was lost in the reference frame due to acquisition process, but it is distributed in other frames.

This chapter presents a survey of the literature on multi-frame super-resolution. We present some classical approaches to the problem and conclude by introducing more recent approaches.

G.1 Frequency Domain Super-resolution

Frequency domain approaches are the earliest attempts to solve the problem of super-resolution [213–217]. They are based on the shifting property of the Fourier transform and on the aliasing relationship between the continuous Fourier transform (CFT) and the discrete Fourier transform (DFT). A system of equations is formulated relating the aliased DFT coefficients of the LR observed images to the samples of the CFT of the unknown HR image. These equations are solved yielding the frequency domain coefficients of the original image, which may then be recovered by IDFT. The formulation of the system of equations requires knowledge of the translational motion between frames with subpixel accuracy.

We present now an overview of frequency domain approaches. Let $f_a(x, y)$ be a real image, still in continuous domain with horizontal coordinate x and vertical coordinate y, and let

$$g_{a,k}(x, y) = f_a(x + h_k, y + v_k) \quad k = 1, \cdots, K \tag{G.1}$$

be the K observations of $f_a(x, y)$ subject to horizontal and vertical translational shifts h_k and v_k respectively. If $F_a(j\Omega_x, j\Omega_y)$ is the Fourier transform of $f(x, y)$, then we have

$$G_{a,k}(j\Omega_x, j\Omega_y) = F_a(j\Omega_x, j\Omega_y) \exp\{j2\pi(h_k\Omega_x + v_k\Omega_y)\} \tag{G.2}$$

as the Fourier transforms of each observation. As the observed images are actually sampled and supposing the sampling periods T_1 and T_2 in the horizontal and vertical directions, respectively, the observed images can be rewritten as

$$g_k(n_1, n_2) = g_{a,k}(n_1T_1, n_2T_2), \tag{G.3}$$

with $n_1 = 1, \cdots, N_1$ and $n_2 = 1, \cdots, N_2$ and N_1 and N_2 being the number vertical and horizontal pixels.

Now we write the DFT of $g_k(n_1, n_2)$ as

$$G_k(m_1, m_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} g_k(n_1, n_2) \exp\left\{-j2\pi\left(\frac{m_1n_1}{N_1} + \frac{m_2n_2}{N_2}\right)\right\}. \tag{G.4}$$

We can relate the spectrum of the sampled observed image with the spectrum of continuous observed images through

$$G_k(m_1, m_2) = \frac{1}{T_1T_2} \sum_{l_1=-\infty}^{\infty} \sum_{l_2=-\infty}^{\infty} G_{a,k}\left(\frac{2\pi}{T_1}\left[\frac{m_1}{N_1} + l_1\right], \frac{2\pi}{T_2}\left[\frac{m_2}{N_2} + l_2\right]\right). \tag{G.5}$$

If we suppose that $F_a(j\Omega_x, j\Omega_y)$ is band-limited, that is, $|F_a(j\Omega_x, j\Omega_y)| = 0$,

$|\Omega_x| > L_1\frac{2\pi}{T_1}$ and $|\Omega_y| > L_2\frac{2\pi}{T_2}$, then each $G_{a,k}(j\Omega_x, j\Omega_y)$ is also band-limited and the infinite summation in (G.5) can be reduced to

$$G_k(m_1, m_2) = \frac{1}{T_1 T_2} \sum_{l_1=-L_1}^{L_1-1} \sum_{l_2=-L_2}^{L_2-1} G_{a,k}\left(\frac{2\pi}{T_1}\left[\frac{m_1}{N_1} + l_1\right], \frac{2\pi}{T_2}\left[\frac{m_2}{N_2} + l_2\right]\right). \quad \text{(G.6)}$$

Now we want to write the DFT coefficients $G_k(m_1, m_2)$ as a function of the spectrum $F_a(j\Omega_x, j\Omega_y)$ of the continuous image. First, we substitute (G.2) on the right-hand side of (G.6) and we have

$$G_k(m_1, m_2) = \frac{1}{T_1 T_2} \sum_{l_1=-L_1}^{L_1-1} \sum_{l_2=-L_2}^{L_2-1} F_a\left(\frac{2\pi}{T_1}\left[\frac{m_1}{N_1} + l_1\right], \frac{2\pi}{T_2}\left[\frac{m_2}{N_2} + l_2\right]\right) \cdot \cdot \exp\left\{j2\pi\left[\frac{h_k}{T_1}\left(\frac{m_1}{N_1} + l_1\right) + \frac{v_k}{T_2}\left(\frac{m_2}{N_2} + l_2\right)\right]\right\} \quad \text{(G.7)}$$

To simplify the notation, we put $\omega_1 = \frac{2\pi}{T_1}\frac{m_1}{N_1}$ and $\omega_2 = \frac{2\pi}{T_2}\frac{m_2}{N_2}$ and divide the spectrum $F_a(\cdot)$ in parts according to the indexes l_1 and l_2. That is, for $l_1 = -L_1$ and $l_2 = -L_2$ we have the first portion which we call $F_{a,1}(\omega_1, \omega_2)$. For $l_1 = L_1 - 1$ and $l_2 = L_2 - 1$ we have the last portion of the spectrum, which we call $F_{a,4L_1L_2}(\omega_1, \omega_2)$. The double summation in (G.7) represents the *aliasing* taking place, when different portions of the continuous image spectrum are mixed, according to the shifts h_k and v_k, due to the sampling process.

Rewriting (G.7) in matrix notation and dividing the continuous image spectrum in parts, we can write the following system of equations

$$\underbrace{\begin{bmatrix} G_1(m_1, m_2) \\ G_2(m_1, m_2) \\ \vdots \\ G_p(m_1, m_2) \end{bmatrix}}_{pM_1 \times M_2} = \frac{1}{T_1 T_2}\mathbf{H} \underbrace{\begin{bmatrix} F_{a,1}(\omega_1, \omega_2) \\ F_{a,2}(\omega_1, \omega_2) \\ \vdots \\ F_{a,4L_1L_2}(\omega_1, \omega_2) \end{bmatrix}}_{4L_1L_2M_1 \times M_2} \quad \text{(G.8)}$$

with

$$[\mathbf{H}]_{k,\mathbf{m},\mathbf{l}} = \exp\left\{j2\pi\left[\frac{h_k}{T_1}\left(\frac{m_1}{N_1} + l_1\right) + \frac{v_k}{T_2}\left(\frac{m_2}{N_2} + l_2\right)\right]\right\}. \quad \text{(G.9)}$$

By solving $\mathbf{G} = \frac{1}{T_1 T_2}\mathbf{H}\mathbf{F}_a$ we can retrieve the spectrum of the original image without *aliasing* and consequently the original image by an IDFT. The shifts between the observations must ensure that the matrix $\mathbf{H}$ is non-singular. The solution is given by

$$\hat{f}(n_1, n_2) = \text{IDFT}\{\mathbf{F}_a\}. \quad \text{(G.10)}$$

Particularly, if the sampling frequency F_s of original image is between $F_{max}/2 \leq$

$F_s < F_{max}$, the aliasing will be confined within the interval $\pm 2\pi$. In this case we will have $L_1 = L_2 = 1$ and the matrix $\mathbf{H}$ becomes square with 4 observations.

G.2 Nonuniform Interpolation Approach

Another very intuitive approach is the interpolation on a nonuniform grid. A sequence of images of the same scene containing translational shifts are first aligned resulting in a combined image composed of samples on a nonuniformly spaced sampling grid. The alignment is obtained by estimating the relative motion, i.e., registration, between frames. These nonuniformly spaced sample points are, then, interpolated and resampled on an HR sampling grid. The final stage is usually a deblurring process, depending on the observation model.

Alam *et al.* [218] utilize a gradient-based registration algorithm to estimate the shifts between the acquired frames and then use a weighted nearest-neighbor approach for placing the frames onto a uniform grid to form a final HR image. Finally, Wiener filtering is applied to reduce effects of blurring and noise caused by the system.

First, each observed frame is modeled as a reference frame $g_{a,0}(x, y)$ with horizontal and vertical shifts as

$$g_{a,k}(x, y) = g_{a,0}(x + h_k, y + v_k). \tag{G.11}$$

An approximation of (G.11) considering the first three terms for the Taylor series expansion is given by

$$g_{a,k}(x, y) \approx g_{a,0}(x, y) + h_k \frac{\partial g_{a,0}(x, y)}{\partial x} + v_k \frac{\partial g_{a,0}(x, y)}{\partial y}. \tag{G.12}$$

The registration parameters h_k and v_k are obtained by solving a least-squares problem considering a discrete version of (G.12) given by

$$\varepsilon_k(h_k, v_k) = \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} \Bigg[g_k(n_1, n_2) - \\ - g_0(n_1, n_2) - h_k \frac{\partial g_0(n_1, n_2)}{\partial n_1} - v_k \frac{\partial g_0(n_1, n_2)}{\partial n_2} \Bigg]^2. \tag{G.13}$$

Once the relative shifts are determined, the frames are placed on the HR grid forming a draft $f_d(n_1, n_2)$ of the HR image. However, some pixels will be missing since they do not have a direct correspondence from LR frames. These missing pixes are filled using a weighted nearest neighbor technique, which consists in averaging the three nearest available pixels giving weights inversely proportional to their distance from the desired point.

Kaltenbacher and Hardie [219] used a similar approach to determine the relative

shifts in LR images. Then, the frequency domain algorithm proposed by Tsai and Huang [213] was used to determine the HR image.

Komatsu *et al.* [220, 221] developed a scheme of acquiring HR images from multiple LR cameras. They employ the block-matching technique to measure relative shifts and utilize the Landweber algorithm [222] to reconstruct an improved resolution image. Suppose we model the nonuniform sampling process as

$$f_d(n_1, n_2) = \mathcal{A}\{f(n_1, n_2)\}, \tag{G.14}$$

where $\mathcal{A}$ is a linear operator representing the nonuniform sampling process, $f(n_1, n_2)$ is the desired HR image and $f_d(n_1, n_2)$ is a version of HR image where the samples are nonuniformly spaced. Since $\mathcal{A}$ is a linear operator, we can use matrix notation and rewrite (G.14) as

$$\mathbf{f}_d = \mathbf{A}\mathbf{f}. \tag{G.15}$$

As mentioned earlier, $\mathbf{f}_d$ is obtained by aligning the LR frames on an HR grid forming a draft of HR image. Since the samples in $\mathbf{f}_d$ are nonuniformly spaced, there might be missing pixels when considering a uniform grid.

The Landweber algorithm described in [220, 221] aims to estimate the HR image by minimizing the cost function

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \ \|\mathbf{f}_d - \mathbf{A}\mathbf{f}\| \tag{G.16}$$

using a gradient-based recursion

$$\hat{\mathbf{f}}_{q+1} = \hat{\mathbf{f}}_q + \mu\mathbf{A}^{\mathrm{T}}(\mathbf{f}_d - \mathbf{A}\mathbf{f}), \tag{G.17}$$

where the index q indicates the refinement level of the estimate.

Following the same idea of interpolation on nonuniform grid, Nguyen and Milanfar [223, 224] proposed a wavelet-based method for super-resolution. As the previous methods, the frames have first to be aligned and placed on a nonuniform grid. Then, the nonuniformly spaced samples are used to estimate the wavelet expansion coefficients of the desired image. Finally, the pixels of the HR image are obtained through the wavelet expansion equation.

For simplicity, we describe here the wavelet interpolation for nonuniformly sampled 1D signals. Based on the same idea, the method is extended to 2D data (i.e. images) in [223, 224].

The wavelet theory [130, 225] states that any function $f(t) \in L^2(\mathbf{R})$ can be decomposed in a set of orthonormal basis functions called *scaling functions* along

with *wavelet functions* through the equation

$$f(t) = \sum_{k \in \mathbf{Z}} a_{J,k} \phi_{J,k}(t) + \sum_{j \geq J} \sum_{k \in \mathbf{Z}} b_{j,k} \psi_{j,k}(t), \tag{G.18}$$

where $\phi_{J,k}(t)$ are the *scaling functions* and $\psi_{j,k}(t)$ are the *wavelet functions*.

The first term on the right-hand side of (G.18) provides a coarse approximation, determined by J, for $f(t)$, whereas the second term adds the "details" to complete the representation of $f(t)$.

The functions $\{\phi(t-k)\}_{k \in \mathbf{Z}}$ form an orthonormal basis for a vector space $\mathcal{V}_0$, whereas their dilatations and translations $\{\phi_{j,k}(t) = 2^{j/2}\phi(2^j t - k)\}_{k \in \mathbf{Z}}$ form an orthonormal basis for $\mathcal{V}_j$. The approximation becomes more accurate as j increases. The difference in successive approximations $z_j(t) = f_{j+1}(t) - f_j(t)$ is a detail signal that lives in a wavelet subspace $\mathcal{W}_j$. In fact, we can decompose the approximation space $\mathcal{V}_{j+1}$ as

$$\mathcal{V}_{j+1} = \mathcal{V}_j \oplus \mathcal{W}_j, \tag{G.19}$$

where the direct sum of subspaces is defined as $H \oplus K = \{\mathbf{u} + \mathbf{v} \,|\, \mathbf{u} \in H, \mathbf{v} \in K\}$.

Finally, any function in $L^2(\mathbf{R})$ can be written as a sum of its approximation at some scale J along with the subsequent detail components at scale J and higher, that is,

$$L^2(\mathbf{R}) = \mathcal{V}_J \oplus \bigoplus_{j \geq J} \mathcal{W}_j, \tag{G.20}$$

which shows a vector-space representation of equation (G.18).

Suppose that we have a function $f(t)$ for which we want to compute N uniformly spaced values at $t = 0, 1, \cdots, N-1$. We are given P nonuniformly sampled data points of $f(t)$ at $t = t_0, t_1, \cdots, t_{P-1}$, $0 \leq t_i \leq N$, where typically $P < N$. We take the unit-time spacing grid (the values of $f(t)$ for $t = 0, 1, \cdots, N-1$) to be the representation of $f(t)$ at the resolution level $\mathcal{V}_0$. Thus, we can write

$$\mathcal{V}_0 = \mathcal{V}_J \oplus \bigoplus_{j=J}^{-1} \mathcal{W}_j, \quad J \leq -1. \tag{G.21}$$

Since the function $\phi(t)$ and $\psi(t)$ have a finite support interval, we can eliminate from (G.18) the summation terms which are zero for a given t_i. Let S_J be the set of shifts of k where the contribution of $\phi(t)$ and $\psi(t)$ are nonzero. Then, we can write

$$f(t_i) = \sum_{k \in S_J} a_{J,k} \phi_{J,k}(t_i) + \sum_{j=J}^{-1} \sum_{k \in S_J} b_{j,k} \psi_{j,k}(t_i), \quad i = 0, \cdots, P-1. \tag{G.22}$$

In vector form, equation (G.22) becomes

$$\mathbf{f} = \mathbf{\Phi}_J \mathbf{a}_J + \sum_{j=J}^{-1} \mathbf{\Psi}_j \mathbf{b}_j, \tag{G.23}$$

where

$$\mathbf{f} = (f(t_i))_{i=0,\cdots,P-1}, \quad \mathbf{a}_J = (a_{J,k})_{k\in S_J}, \quad \mathbf{b}_j = (b_{j,k})_{k\in S_j},$$
$$\mathbf{\Phi}_J = (\phi_{J,k}(t_i))_{i=0,\cdots,P-1}^{k\in S_J}, \quad \mathbf{\Psi}_j = (\psi_{j,k}(t_i))_{i=0,\cdots,P-1}^{k\in S_j}.$$

To construct $\mathbf{\Phi}_J$ and $\mathbf{\Psi}_j$, we need to know the basis functions values at sampling points $\{t_i\}$. For most wavelet bases, there are no closed-form expressions for basis functions. However, basis function values at dyadic points can be calculated efficiently by recursion [223, 224].

From equation (G.23), we can approximate the coarse-scale coefficients $\mathbf{a}_J$ by ignoring the detail components and considering just

$$\mathbf{f} \approx \mathbf{\Phi}_J \mathbf{a}_J. \tag{G.24}$$

Choosing J appropriately, we can solve the above system in a least-squares sense, that is,

$$\hat{\mathbf{a}}_J = (\mathbf{\Phi}_J^{\mathrm{T}} \mathbf{\Phi}_J + \lambda \mathbf{I})^{-1} \mathbf{\Phi}_J^{\mathrm{T}} \mathbf{f}, \tag{G.25}$$

for some regularization parameter λ. The least-squares estimate $\hat{\mathbf{a}}_J$ yields a coarse-scale approximation of $\mathbf{f}$. The remaining details can be obtained by

$$\begin{aligned} \mathbf{z}_J &= \mathbf{f} - \mathbf{\Phi}_J \hat{\mathbf{a}}_J \\ &\approx \mathbf{\Psi}_J \mathbf{b}_J, \end{aligned} \tag{G.26}$$

which, being solved in the least-squares sense, yields $\hat{\mathbf{b}}_J$. The desired vales of $f(t)$ at the HR grid points $t = 0, 1, \cdots, N-1$ can then be computed using the estimated coefficients:

$$f(t) \approx \sum_{k\in S_J} \hat{a}_{J,k} \phi_{J,k}(t_i) + \sum_{k\in S_J} \hat{b}_{J,k} \psi_{J,k}(t_i), \quad t = 0, \cdots, N-1. \tag{G.27}$$

G.3 POCS: Projection onto Convex Sets

According to the method of POCS [67, 226], incorporating a priori knowledge into the solution can be interpreted as restricting the solution to be a member of a closed convex set C_j that is defined as a set of vectors, or images, which satisfy a particular property. If the constraint sets have a nonempty intersection, then a

solution that belongs to the intersection $C_s = \bigcap_{j=1}^{J} C_j$, which is also a convex set, can be found by alternating projections onto these convex sets. Indeed, any solution in the intersection set is consistent with the a priori constraints and therefore it is a feasible solution. The method of POCS can be applied to find an image which belongs in the intersection by the recursion

$$\hat{f}_{q+1}(n_1, n_2) = P_J P_{J-1} \cdots P_2 P_1 \hat{f}_q(n_1, n_2), \tag{G.28}$$

where, for $q = 0$, $\hat{f}_0$ is an arbitrary starting estimate, and P_j is the projection operator which projects an arbitrary signal $\hat{f}$ onto the closed, convex sets, C_j $(j = 1, ..., J)$. As we are talking about sets and projection, it is interesting to use vector notation and rewrite (G.28) as

$$\hat{\mathbf{f}}_{q+1} = P_J P_{J-1} \cdots P_2 P_1 \hat{\mathbf{f}}_q. \tag{G.29}$$

Stark and Oskoui [226] proposed a method to solve the problem of super-resolution using POCS. As convex sets, they use data constraint sets C_j, which assure that the estimated HR image is consistent with the observed data, along with additional constraints that are imposed from prior knowledge concerning the HR image, such as amplitude constraint, energy constraint and reference-image constraint.

Tekalp *et al.* [227] extended the above approach to take into account the presence of both sensor blurring and observation noise. Patti *et al.* [228] adopted an alternative formulation which also takes into account the nonzero aperture time of the sensor. Later in [229], Patti *et al.* improved his POC-based method by modifying the constraint sets to reduce the amount of edge ringing present in the HR image estimate. Through this modification, a better regularization of inversion process was achieved.

Altunbasak *et al.* [230] studied the application of a super-resolution method to MPEG-coded video. The algorithm is a motion-compensated, transform-domain (DCT) super-resolution procedure that incorporates the transform-domain quantization information.

G.4 Regularized Approaches

G.4.1 Deterministic Approach

Peleg *et al.* [231] proposed an early simple scheme to super-resolution. Given a set of K LR observed images $g_k(n_1, n_2)$, the algorithm starts with an initial guess for the desired HR image $\hat{f}(n_1, n_2)$, and simulates the imaging process to get a initial set of simulated LR images $\hat{g}_k(n_1, n_2)$. The error between this set and the set of observed

images is defined as

$$\varepsilon = \sum_{k} \sum_{n_1,n_2} \left| \hat{g}_k(n_1, n_2) - g_k(n_1, n_2) \right|. \tag{G.30}$$

All pixels of the guess are examined. If the current gray level is $\hat{f}(n_1, n_2) = l$, three possible updates are considered: $\{l-1, l, l+1\}$. The value which minimizes the error in (G.30) is chosen as update for $\hat{f}(n_1, n_2)$. The process is continued iteratively until no further improvement can be obtained or until the maximum number of allowed iterations is reached.

Irani and Peleg [232] modeled the acquisition process as blurred LR observations of an HR image subject to rotations and translations. The iterative back-projection (IBP) SR reconstruction approach, similar to the back projection used in tomography, was developed. In this approach, the HR image is estimated by back projecting the error (difference) between simulated LR images via imaging blur and the observed LR images. This process is repeated iteratively to minimize the energy of the error.

Hardie *et al.* [233] also considered blurred LR observations of an HR image subject to rotations and translations. However, the HR image was estimated via gradient descent optimization. The cost function used considered data consistency (or data matching term), which ensures that the solution agrees to the observed images, and a regularization term, which is minimized when the estimated HR image is smooth.

Elad and Feuer [234, 235] proposed a method based on adaptive filtering theory for super-resolution restoration of continuous image sequences. The proposed methodology suggests least squares (LS) estimators which adapt in time, based on adaptive least mean squares (LMS) or recursive least squares (RLS) filters. The observed images are assumed to be driven from a source image through the following equation

$$\mathbf{g}_k = \mathbf{RHM}_k\mathbf{f} + \mathbf{v}_k, \tag{G.31}$$

where $\mathbf{g}_k$ are indexed vectors representing the observed images with the columns lexicographically ordered, $\mathbf{R}$ represents the decimation operator, $\mathbf{H}$ represents the blur matrix, $\mathbf{M}_k$ represents the geometric warp matrix, $\mathbf{f}$ is a vector representing the desired HR image with the columns lexicographically ordered and $\mathbf{v}_k$ represents an additive random noise.

The decimation, the blur and the warp matrices are assumed to be known. The super-resolution estimated image can be obtained by minimizing

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \left\{ \|\mathbf{Sf}\|^2 + \sum_{k} \|\mathbf{g}_k - \mathbf{RHM}_k\mathbf{f}\|^2 \right\} \tag{G.32}$$

where the first term penalizes for the nonsmoothness of the obtained estimation.

This term is a regularization mechanism, which algebraically regularizes the problem so that it has a unique optimal solution, and physically forces the estimation result to satisfy *a priori* knowledge about the smoothness of the ideal desired image.

Elad and Feuer [234, 235] derived two algorithms: a pseudo-RLS, which uses the steepest descent method to avoid matrix inversions, and an LMS algorithm, which is obtained by a simplification in the pseudo-RLS algorithm using instantaneous approximations.

Later in [115], Elad and Hel-Or exploited previous results to develop a new efficient super-resolution reconstruction algorithm which separates the treatment into de-blurring and measurements fusion. The fusion part is a simple noniterative algorithm.

In order to improve the convergence of the conjugate gradient method, Nguyen *et al.* [236] proposed efficient block circulant preconditioners for solving the Tikhonov-regularized super-resolution problem. Preconditioning is a technique used to transform the original system into one with the same solution, but which can be solved by the iterative solver more quickly.

G.4.2 Probabilistic Methods

Since super-resolution is an ill-posed inverse problem, techniques which are capable of including *a priori* constraints are well suited to this application. Bayesian methods inherently include *a-priori* constraints in the form of prior probability density functions. The Bayesian approach is sometimes used as a synonymous of maximum *a-posteriori* (MAP) estimation [237].

Given the general observation model

$$\mathbf{g}_k = \mathbf{W}_k \mathbf{f} + \mathbf{v}_k, \tag{G.33}$$

where $\mathbf{W}_k$ is a matrix which performs decimation, blur and geometric warping. The MAP estimator of $\mathbf{f}$ maximizes the a posteriori PDF $p(\mathbf{f}|\mathbf{g}_k)$ with respect to $\mathbf{f}$ [67]

$$\hat{\mathbf{f}} = \arg\max_{\mathbf{f}} \; p(\mathbf{f}|\mathbf{g}_1, \cdots, \mathbf{g}_K). \tag{G.34}$$

Applying Bayes' rule yields,

$$\hat{\mathbf{f}} = \arg\max_{\mathbf{f}} \; \frac{p(\mathbf{g}_1, \cdots, \mathbf{g}_K|\mathbf{f})p(\mathbf{f})}{p(\mathbf{g}_1, \cdots, \mathbf{g}_K)} \tag{G.35}$$

and since the maximum is independent of $\mathbf{g}_k$ we have,

$$\hat{\mathbf{f}} = \arg\max_{\mathbf{f}} \; p(\mathbf{g}_1, \cdots, \mathbf{g}_K|\mathbf{f})p(\mathbf{f}). \tag{G.36}$$

Since the logarithm is a monotonic increasing function, this is equivalent to finding,

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \left\{ -\log p(\mathbf{g}_1, \cdots, \mathbf{g}_K|\mathbf{f}) - \log p(\mathbf{f}) \right\}, \tag{G.37}$$

where $\log p(\mathbf{g}_1, \cdots, \mathbf{g}_K|\mathbf{f})$ is the log-likelihood function and $\log p(\mathbf{f})$ is the log of the *a-priori* density of $\mathbf{f}$. Both the *a-priori* image model $p(\mathbf{f})$ and the conditional probability $p(\mathbf{g}_1, \cdots, \mathbf{g}_K|\mathbf{f})$ will be defined by *a-priori* knowledge concerning the HR image $\mathbf{f}$ and the statistical information of noise. Since MAP optimization in (G.37) includes *a-priori* constraints (prior knowledge represented by $p(\mathbf{f})$) essentially, it provides regularized (stable) SR estimates effectively. Markov random field (MRF) prior is often adopted for prior image model since it provides a powerful method for image prior modeling [67].

Schultz and Stevenson [238] developed a super-resolution image reconstruction algorithm using a Huber-Markov Random Field (HMRF) prior. Schultz considered HMRF a more reasonable prior assumption since the digitized data is piece-wise smooth, i.e., image data consists of smooth regions, with these regions separated by discontinuities. The Huber-Markov random field (HMRF) model is a Gibbs prior that represents piecewise smooth data, with the probability density defined as

$$p(\mathbf{f}) = \frac{1}{Z} \exp\left\{ -\frac{1}{2\beta} \sum_{c \in C} \rho\left(\mathbf{d}_c^{\mathrm{T}} \mathbf{f}\right) \right\}, \tag{G.38}$$

where Z is a normalizing constant, β is the "temperature" parameter for the Gibbs prior, c is a local group of pixels contained within the set of all image cliques C, $\rho(\cdot)$ is the energy function and $\mathbf{d}_c$ is a spatial activity measure within the clique c, with a small value in smooth image locations and a large value at edges.

The likelihood of edges in the data is controlled by the Huber edge penalty function

$$\rho(x) = \begin{cases} x^2, & |x| \le \alpha \\ 2\alpha\,|x| - \alpha^2, & |x| > \alpha, \end{cases} \tag{G.39}$$

where α is a threshold parameter separating the quadratic and linear regions. A quadratic edge penalty,

$$\rho(x) = x^2 \tag{G.40}$$

characterizes the Gauss-Markov image model. Edges are severely penalized by the quadratic function, making discontinuities within the Gaussian image model unlikely. The threshold parameter α controls the size of discontinuities modeled by the prior by providing a less severe edge penalty.

A MAP framework for the joint estimation of image registration parameters and the HR image was presented by Hardie *et al.* in [239]. The registration parameters,

horizontal and vertical shifts in this case, are iteratively updated along with the HR image in a cyclic optimization procedure.

Farsiu *et al.* [108] proposed an approach to super-resolution reconstruction using $\ell_1$norm minimization and robust regularization based on a bilateral prior to deal with different data and noise models. The method is claimed to be robust to errors in motion and blur estimation and to result in images with sharp edges. Farsiu *et al.* considered the popular acquisition model defined as

$$\mathbf{g}_k = \mathbf{RHM}_k\mathbf{f} + \mathbf{v}_k, \tag{G.41}$$

where $\mathbf{g}_k$ are indexed vectors representing the observed images with the columns lexicographically ordered, $\mathbf{R}$ represents the decimation operator, $\mathbf{H}$ represents the blur matrix, $\mathbf{M}_k$ represents the geometric warp matrix, $\mathbf{f}$ is a vector representing the desired HR image with the columns lexicographically ordered and $\mathbf{v}_k$ represents an additive random noise.

The problem is broken in two steps: finding a blurred HR image $\mathbf{z} = \mathbf{Hf}$, called fusion step, and estimating the deblurred HR image $\mathbf{f}$ from $\mathbf{z}$, called deblurring step. The fusion step is performed by minimizing

$$J = \min_{\mathbf{z}} \left\{ \sum_k \|\mathbf{RM}_k\mathbf{z} - \mathbf{g}_k\|_1 \right\}. \tag{G.42}$$

Note that the order of $\mathbf{H}$ and $\mathbf{M}_k$ was changed. It turns out that $\mathbf{H}$ and $\mathbf{M}_k$ are block-circulant matrices since Farsiu *et al.* [108] considered only translational motion. And since they are block-circulant matrices, they commute [116].

An approximation of the gradient of (G.42) is given by

$$\nabla_{\mathbf{z}} J = \sum_k \mathbf{M}_k^{\mathrm{T}}\mathbf{R}^{\mathrm{T}} \operatorname{sign}(\mathbf{RM}_k\mathbf{z} - \mathbf{g}_k), \tag{G.43}$$

The minimization (G.42) is done by the noniterative Shift-and-Add algorithm, which is described next: the term $\mathbf{M}_k^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}$ copies the values from the LR grid to the HR grid after proper shifting and zero filling, whereas $\mathbf{RM}_k$ copies a selected set of pixels in HR grid back on the LR grid. Neither of these two operations changes the pixel values. Therefore, each element of $\mathbf{g}_1$, which corresponds to one element in $\mathbf{z}$, is the aggregate of the effects of all LR frames. The effect of each frame in $\mathbf{z}$ has one of the following three forms:

1. addition of zero, which results from zero filling;

2. addition of +1, which means a pixel in $\mathbf{z}$ was larger than the corresponding contributing pixel from frame $\mathbf{g}_k$;

3. addition of -1, which means a pixel in $\mathbf{z}$ was smaller than the corresponding contributing pixel from frame $\mathbf{g}_k$.

The deblurring step is obtained by minimizing

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \left\{ \|\mathbf{H}\mathbf{f} - \mathbf{z}\|_1 + \lambda \underbrace{\sum_{l=-P}^{P} \sum_{m=0}^{P}}_{l+m\geq 0} \alpha^{|m|+|l|} \left\| \mathbf{f} - \mathbf{S}_x^l \mathbf{S}_y^m \mathbf{f} \right\|_1 \right\}, \tag{G.44}$$

where the matrices $\mathbf{S}_x^l$ and $\mathbf{S}_y^m$ perform shift by l and m pixels the image $\mathbf{f}$ in the horizontal and vertical directions respectively. The scalar weight $0 < \alpha < 1$ is applied to give a spatially decaying effect to the summation of the regularization terms. The regularization term on the right-hand side of (G.44) is referred to as Bilateral Total Variation (BTV).

G.5 Super-resolution without Motion Estimation

This section describes some new multi-frame super-resolution algorithms which do not require motion estimation. Classic super-resolution techniques strongly rely on the availability of accurate motion estimation for this fusion task. When motion is inaccurately estimated, as often happens for local motion fields, annoying artifacts appear in the super-resolved outcome [84].

Additionally, the need for precise motion estimates in conventional super-resolution has limited its applicability to only video sequences with relatively simple motion such as global translational or affine displacements. The goal of the techniques presented in this section is to overcome this drawback and expand the applicability of super-resolution to a broader range of video sequences [84, 85, 240].

G.5.1 Generalizing Nonlocal-Means to Super-resolution Reconstruction

Protter *et al.* [84] proposed a super-resolution scheme based on a generalization of the Nonlocal-Means (NLM) algorithm. The idea was to overcome the need of explicit motion estimation. The NLM filter is based on the assumption that image content is likely to repeat itself within some neighborhood. Therefore, denoising of each pixel is done by averaging all pixels in its neighborhood, that is,

$$\hat{f}(\mathbf{n}) = \frac{\sum_{\mathbf{m}\in\mathcal{N}(\mathbf{n})} w(\mathbf{n},\mathbf{m}) g(\mathbf{m})}{\sum_{\mathbf{m}\in\mathcal{N}(\mathbf{n})} w(\mathbf{n},\mathbf{m})}, \quad \mathbf{n} = [n_1, n_2]^{\mathrm{T}}, \quad \mathbf{m} = [m_1, m_2]^{\mathrm{T}}, \tag{G.45}$$

where $\hat{f}(\mathbf{n})$ is the filtered image, $y(\mathbf{m})$ is the input image, $w(\mathbf{n},\mathbf{m})$ are the filter weights and $\mathcal{N}(\mathbf{n})$ stands for the neighborhood of $\mathbf{n}$.

Defining $\mathbf{P_n}$ as an operator that extracts a patch of a fixed and predetermined size (say $q \times q$ pixels) from an image around the position $\mathbf{n}$, and lexicographically ordering the columns of the image into a vector $\mathbf{g}$, the expression $\mathbf{P_n g}$ results with a vector of length q^2 being the extracted patch. Thus, the NLM weights are given by

$$w(\mathbf{n},\mathbf{m}) = \exp\left\{-\frac{\|\mathbf{P_n g} - \mathbf{P_m g}\|^2}{2\sigma^2}\right\} \cdot \gamma\left(\|\mathbf{n}-\mathbf{m}\|\right), \tag{G.46}$$

where the function $\gamma(\cdot)$ takes the geometric distance into account, and as such, it is monotonically nonincreasing. In other words, the weights for the NLM filter are computed based both on radiometric (gray-level) proximity and geometric proximity between the pixels.

The NLM filter described in (G.45) and (G.46) can be deduced by minimizing a properly defined penalty function [84, 241, 242]

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \left\{\frac{1}{2}\sum_{\mathbf{n}}\sum_{\mathbf{m}\in\mathcal{N}(\mathbf{n})} w(\mathbf{n},\mathbf{m}) \cdot \|\mathbf{P_n f} - \mathbf{P_m g}\|^2\right\}. \tag{G.47}$$

Before turning to super-resolution, the temporal axis can be introduced into the penalty function, so as to process a sequence of images and not just a single one. Thus, we get

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \left\{\frac{1}{2}\sum_{\mathbf{n}}\sum_{k}\sum_{\mathbf{m}\in\mathcal{N}(\mathbf{n})} w(\mathbf{n},\mathbf{m},k) \cdot \|\mathbf{P_n f} - \mathbf{P_m g}_k\|^2\right\}. \tag{G.48}$$

Considering the problem of super-resolution, the LR observed images $\mathbf{g}_k$ are blurred and decimated versions of the HR image $\mathbf{f}$. We have, therefore, to apply a blurring operator and a decimation operator into HR image, that is, $\mathbf{RHf}$. Protter *et al.* [84] divided the solution into two steps: defining $\mathbf{z} = \mathbf{Hf}$, the first step is to estimate $\mathbf{z}$ from the measurements $\mathbf{g}_k$. This is done by minimizing the energy function

$$\hat{\mathbf{z}} = \arg\min_{\mathbf{z}} \left\{\frac{1}{2}\sum_{\mathbf{n}}\sum_{k}\sum_{\mathbf{m}\in\mathcal{N}(\mathbf{n})} w(\mathbf{n},\mathbf{m},k) \cdot \left\|\mathbf{R}_p\mathbf{P}_{\mathbf{n}}^H\mathbf{z} - \mathbf{P}_{\mathbf{m}}^L\mathbf{g}_k\right\|^2\right\}, \tag{G.49}$$

where $\mathbf{R}_p$ is a decimation matrix applied to the patch, $\mathbf{P}_{\mathbf{n}}^H$ is the image patch extraction operator at high resolution around the position $\mathbf{n}$ and $\mathbf{P}_{\mathbf{m}}^L$ is the image patch extraction operator at low resolution around the position $\mathbf{m}$. The idea

of breaking the super-resolution task into two parts (fusing the inputs and then deblurring) has been suggested previously in [108, 115].

Once the fusion step is completed, the deblurring step, which results in the HR image, is done by minimizing

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \left\{ \|\mathbf{z} - \mathbf{H}\mathbf{f}\| + \lambda \mathrm{TV}(\mathbf{f}) \right\}, \tag{G.50}$$

where $\lambda \mathrm{TV}(\mathbf{f})$ is the Total Variation (TV) regularization [20].

G.5.2 Super-resolution through Multidimensional Kernel Regression

Takeda *et al.* [85] proposed a super-resolution algorithm, based on 3D Steering Kernel Regression, to handle complex and arbitrary motion on general sequences, while avoiding explicit (subpixel-accurate) motion estimation. First, we describe the classical kernel regression, and then we extend the idea to the 3D Steering Kernel Regression technique. The Kernel Regression framework defines its data model as

$$g_i = f(\mathbf{n}_i) + v_i, \quad i = 1, \cdots, N \quad \mathbf{n} = [n_1, n_2]^{\mathrm{T}}, \tag{G.51}$$

where g_i is a noisy sample at position $\mathbf{n}_i$, $f(\cdot)$ is the *regression function* to be estimated and v_i is random noise. While the particular form of $f(\cdot)$ may remain unspecified, we can develop a generic local expansion of the function about a sampling point $\mathbf{n}_i$. Specifically, if $\mathbf{n}$ is near the sample $\mathbf{n}_i$, the Taylor series are given by

$$\begin{aligned} f(\mathbf{n}_i) &\approx f(\mathbf{n}) + [\nabla f(\mathbf{n})]^{\mathrm{T}} (\mathbf{n}_i - \mathbf{n}) + \frac{1}{2}(\mathbf{n}_i - \mathbf{n})^{\mathrm{T}} [\mathcal{H} f(\mathbf{n})] (\mathbf{n}_i - \mathbf{n}) + \cdots \\ &\approx \beta_0 + \boldsymbol{\beta}_1^{\mathrm{T}} (\mathbf{n}_i - \mathbf{n}) + \boldsymbol{\beta}_2^{\mathrm{T}} \operatorname{vech}\left[(\mathbf{n}_i - \mathbf{n})(\mathbf{n}_i - \mathbf{n})^{\mathrm{T}} \right] + \cdots, \end{aligned} \tag{G.52}$$

where ∇ and $\mathcal{H}$ are the gradient and Hessian operators respectively, and $\operatorname{vech}(\cdot)$ is the half-vectorization operator that lexicographically orders the lower triangular portion of a symmetric matrix into a column-stacked vector. Furthermore, β_0 is $f(\mathbf{n})$, which is the signal (or pixel) value of interest, and the vectors $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ are

$$\begin{aligned} \boldsymbol{\beta}_1 &= \left[\frac{\partial f(\mathbf{n})}{\partial n_1}, \quad \frac{\partial f(\mathbf{n})}{\partial n_2} \right]^{\mathrm{T}} \\ \boldsymbol{\beta}_2 &= \left[\frac{\partial^2 f(\mathbf{n})}{\partial^2 n_1^2}, \quad \frac{\partial^2 f(\mathbf{n})}{\partial n_1 \partial n_2}, \quad \frac{\partial^2 f(\mathbf{n})}{\partial n_2^2} \right]^{\mathrm{T}}. \end{aligned} \tag{G.53}$$

Since this approach is based on local signal representations, we give the nearby samples higher weights than samples farther away. A (weighted) least-square formu-

lation of the fitting problem capturing this idea is

$$\min_{\{\beta_j\}_{j=1}^{P}} \sum_{i=1}^{N} \Bigg[g_i - \beta_0 - \boldsymbol{\beta}_1^{\mathrm{T}}(\mathbf{n}_i - \mathbf{n}) \\ - \boldsymbol{\beta}_2^{\mathrm{T}} \operatorname{vech}\left[(\mathbf{n}_i - \mathbf{n})(\mathbf{n}_i - \mathbf{n})^{\mathrm{T}}\right] - \cdots \Bigg] Q_{\mathbf{H}_i}(\mathbf{n}_i - \mathbf{n}), \quad \text{(G.54)}$$

with

$$Q_{\mathbf{H}_i}(\mathbf{n}_i - \mathbf{n}) = \frac{1}{\det(\mathbf{H}_i)} q(\mathbf{H}_i^{-1}(\mathbf{n}_i - \mathbf{n})), \quad \text{(G.55)}$$

where P is the regression order, $q(\cdot)$ is the kernel function (a radially symmetric function such as a Gaussian), and $\mathbf{H}_i$ is the smoothing matrix which dictates the "footprint" of the kernel function. The least-squares solution is given by

$$\mathbf{b} = (\mathbf{A}^{\mathrm{T}}\mathbf{Q}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{Q}\mathbf{g}, \quad \text{(G.56)}$$

where

$$\begin{aligned} \mathbf{g} &= [g_1, g_2, \cdots, g_N]^{\mathrm{T}} \\ \mathbf{b} &= [\beta_0, \boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_P]^{\mathrm{T}} \\ \mathbf{Q} &= \operatorname{diag}[Q_{\mathbf{H}_1}(\mathbf{n}_1 - \mathbf{n}), \cdots, Q_{\mathbf{H}_P}(\mathbf{n}_P - \mathbf{n})] \\ \mathbf{A} &= \begin{bmatrix} 1 & (\mathbf{n}_1 - \mathbf{n})^{\mathrm{T}} & \operatorname{vech}^{\mathrm{T}}[(\mathbf{n}_1 - \mathbf{n})(\mathbf{n}_1 - \mathbf{n})^{\mathrm{T}}] & \cdots \\ 1 & (\mathbf{n}_2 - \mathbf{n})^{\mathrm{T}} & \operatorname{vech}^{\mathrm{T}}[(\mathbf{n}_2 - \mathbf{n})(\mathbf{n}_2 - \mathbf{n})^{\mathrm{T}}] & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ 1 & (\mathbf{n}_P - \mathbf{n})^{\mathrm{T}} & \operatorname{vech}^{\mathrm{T}}[(\mathbf{n}_P - \mathbf{n})(\mathbf{n}_P - \mathbf{n})^{\mathrm{T}}] & \cdots \end{bmatrix}. \end{aligned} \quad \text{(G.57)}$$

The steering kernel framework is based on the idea of robustly obtaining local signal structures by analyzing the radiometric (pixel value) differences locally, and feeding this structure information to the kernel function in order to affect its shape and size. Thus, the smoothing matrix $\mathbf{H}_i$ is given by

$$\mathbf{H}_i = h\mathbf{C}_i^{-1/2}, \quad \text{(G.58)}$$

where $\mathbf{C}_i$ is estimated as the local covariance matrix of the neighborhood spatial gradient vectors. In other words, the steering kernel regression is adapted to local signatures of the data.

We can add a time coordinate and extend the steering kernel framework to

$$g_i = f(\mathbf{n}_i) + v_i, \quad i = 1, \cdots, N \quad \mathbf{n} = [n_1, n_2, k]^{\mathrm{T}}, \quad \text{(G.59)}$$

where the additional coordinate k corresponds to the k-th video frame. Following a

similar procedure to the one presented above, we can derive all the equation with the extra coordinate [85]. This results in the so-called 3D steering kernel regression.

The extension of 3D steering kernel for super-resolution [85] is done by first interpolating or upscaling using some reasonably effective low-complexity method (say the "classic" KR method [243]) to yield what is called a *pilot* initial estimate. The pilot estimate is a video sequence with higher resolution than the observed sequence.

The next step is the estimation of the gradients $\hat{\boldsymbol{\beta}}_1^{(0)}$ by (G.56) and the smoothing matrices $\mathbf{H}_i^{(0)}$ by (G.58). With $\mathbf{H}_i^{(0)}$, the pixels' values of the HR image are updated and the gradients are recalculated by (G.56) resulting in $\hat{\boldsymbol{\beta}}_1^{(1)}$. Then, with the new gradients, the smoothing matrices are updated to $\mathbf{H}_i^{(1)}$. The iteration continues until the convergence is reached.

Appendix H

Generalized Gaussian distribution

The Generalized Gaussian distribution is a flexible distribution defined by

$$p(x;\mu,\sigma,\phi) = \frac{1}{2\Gamma(1+1/\phi)A(\phi,\sigma)} \exp\left\{-\left|\frac{x-\mu}{A(\phi,\sigma)}\right|^{\phi}\right\}, \tag{H.1}$$

where

$$A(\phi,\sigma) = \left[\frac{\sigma^2\Gamma(1/\phi)}{\Gamma(3/\phi)}\right]^{1/2}, \tag{H.2}$$

x is a random variable and ϕ a parameter that controls the shape of the distribution independently of its mean or variance. For $\phi = 2$ we have the classical Gaussian, for $\phi = 1$ the Laplacian distribution and so forth. Figure H.1 shows examples of Generalized Gaussian probability density functions for several values of ϕ.

Estimating the shape parameter ϕ is not, in general, an easy task. Some examples of methods can be found in [244–246]. We will use the procedure by Domínguez-Molina *et al.* [246], for which we give a simple description.

First, two functions are defined:

$$M(\phi) = \frac{\Gamma^2(2/\phi)}{\Gamma(1/\phi)\,\Gamma(3/\phi)} \tag{H.3}$$

referred to as *generalized Gaussian function ratio* reciprocal and

$$\hat{M}(x) = \frac{\left(\frac{1}{N}\sum_{n=1}^{N}|x_n-\mu|\right)^2}{\frac{1}{N}\sum_{n=1}^{N}|x_n-\mu|^2}, \tag{H.4}$$

referred to as *sampled generalized Gaussian function ratio* reciprocal.

First, $\hat{M}(x)$ is calculated from the sample through (H.4). Then, we set $M(\phi) = \hat{M}(x)$ and solve (H.3). Because the latter expression does not have a closed-form solution, we can use table lookup/interpolation to obtain the shape parameter ϕ. Proofs and details can be found in [246].

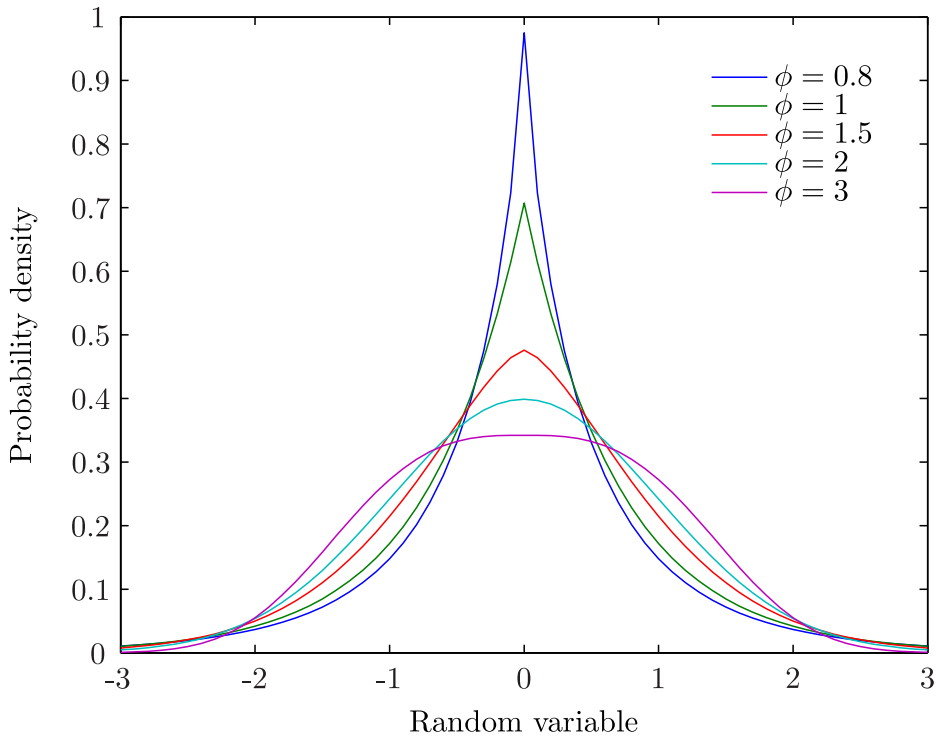


Figure H.1: Example of Generalized Gaussian probability density functions for several values of the shape parameter ϕ

Appendix I

Full-size images

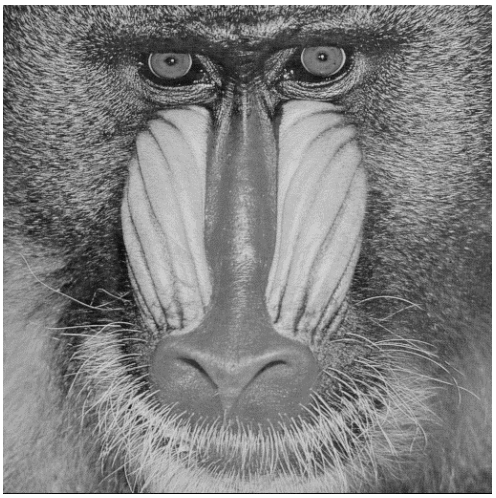
I.1 Classical image dataset

cameraman512.png

lena512.png

barbara512.png

mandril12.png



I.2 Kodak image dataset

kodim01.png

kodim02.png

kodim03.png

kodim04.png

kodim05.png

kodim06.png

kodim07.png

kodim08.png

kodim09.png

kodim10.png

kodim11.png

kodim12.png

kodim13.png

kodim14.png

kodim15.png

kodim16.png

kodim17.png

kodim18.png

kodim19.png

kodim20.png

kodim21.png

kodim22.png

kodim23.png

kodim24.png

I.3 Movies

carphone.png

foreman.png

miss-america.png

suzie.png

Appendix J

Deblurring via Bilateral Total Variation (BTV)

Based on the spirit of TV criterion and a related technique called the bilateral filter [247], Farsiu *et al.* [108] proposed a generalization called bilateral TV. The regularizing function looks like

$$\varphi_{BTV}(\mathbf{f}) = \underbrace{\sum_{l=-P}^{P}\sum_{m=0}^{P}}_{l+m\geq 0} \alpha^{|m|+|l|} \left\|\mathbf{f} - \mathbf{S}_x^l \mathbf{S}_y^m \mathbf{f}\right\|_1, \tag{J.1}$$

where matrices $\mathbf{S}_x^l$ and $\mathbf{S}_y^m$ shift $\mathbf{f}$ by l and m pixels in horizontal and vertical directions respectively. The scalar weight α, $0 < \alpha < 1$ is applied to give a spatially decaying effect to the summation of the regularization terms.

If we limit m and l to the two cases of $m = 1$, $l = 0$ and $m = 0$, $l = 1$ with $\alpha = 1$, and define operators $\mathbf{D}_x$ and $\mathbf{D}_y$ as representatives of the first derivative ($\mathbf{D}_x = \mathbf{I} - \mathbf{S}_x$ and $\mathbf{D}_y = \mathbf{I} - \mathbf{S}_y$) then (J.1) results in

$$\varphi_{BTV}(\mathbf{f}) = \|\mathbf{D}_x\mathbf{f}\|_1 + \|\mathbf{D}_y\mathbf{f}\|_1, \tag{J.2}$$

which is exactly the anisotropic Total Variation regularization, see Chapter 3 and Appendix B.

Referências Bibliográficas

[1] BERTERO, M., BOCCACCI, P. *Introduction to Inverse Problems in Imaging*. Bristol, UK, IOP Pub., 1998. doi:10.1201/9781439822067. 1, 7, 49, 125

[2] PIPA, D. R., DA SILVA, E. A. B., PAGLIARI, C. L., et al. "Recursive Algorithms for Bias and Gain Nonuniformity Correction in Infrared Videos", *Image Processing, IEEE Transactions on*, v. 21, n. 12, 2012. ISSN: 1057-7149. doi:10.1109/TIP.2012.2218820. 5, 41, 46, 47

[3] PIPA, D. R., CHAN, S. H., NGUYEN, T. Q. "Directional decomposition based total variation image restoration". In: *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pp. 1558–1562, Bucharest, Romania, ago. 2012. Disponível em: <http://www.eurasip.org/Proceedings/Eusipco/Eusipco2012/Conference/papers/1569583799.pdf>. 5, 25, 43, 48, 49

[4] PIPA, D. R. *Correção de Ruído de Padrão Fixo em Vídeo Infravermelho*. Tese de Mestrado, COPPE/UFRJ – Universidade Federal do Rio de Janeiro, 2008. Disponível em: <http://www02.lps.ufrj.br/~eduardo/eduardo_oficial/teses/daniel-pipa-mestrado.pdf>. 6, 46, 158, 159

[5] JAIN, A. K. *Fundamentals of digital image processing*. Upper Saddle River, NJ, USA, Prentice-Hall, Inc., 1989. ISBN: 0-13-336165-9. 7, 9, 14

[6] FARIA, S. M. M. *Very Low Bit Rate Video Coding using Geometric Transform Motion Compensation*. Tese de Doutorado, University of Essex, 1996. Disponível em: <http://www.co.it.pt/si/sipub/sf96dt01.pdf>. 7

[7] STRANG, G., NGUYEN, T. *Wavelets and Filter Banks*. Wellesley, MA, Wellesley-Cambridge Press, 1996. ISBN: 978-096140887-9. 8

[8] UNSER, M. "Splines: a perfect fit for signal and image processing", *Signal Processing Magazine, IEEE*, v. 16, n. 6, pp. 22–38, nov. 1999. ISSN: 1053-5888. doi:10.1109/79.799930. 8, 14

[9] PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T., et al. *Numerical Recipes: The Art of Scientific Computing*. 3 ed. New York, Cambridge University Press, 2007. ISBN: 978-052188068-8. 13

[10] SIMONCELLI, E. "Statistical models for images: compression, restoration and synthesis". In: *Signals, Systems amp; Computers, 1997. Conference Record of the Thirty-First Asilomar Conference on*, v. 1, pp. 673–678, nov. 1997. doi:10.1109/ACSSC.1997.680530. 14

[11] TORRALBA, A., OLIVA, A. "Statistics of natural image categories", *Network: Computation in Neural Systems*, v. 14, n. 3, pp. 391–412, ago. 2003. doi:10.1088/0954-898X_14_3_302. Disponível em: <http://cvcl.mit.edu/papers/TorralbaOliva03.pdf>.

[12] CHAN, T. F., SHEN, J. *Image processing and analysis*. Philadelphia, SIAM - Society for Industrial and Applied Mathematics, 2005. ISBN: 978-0898715897. 16

[13] HYVÄRINEN, A., HURRI, J., , et al. *Natural Image Statistics: A probabilistic approach to early computational vision*, v. 39. New York, Springer-Verlag, 2009. ISBN: 978-1-84882-490-4. Disponível em: <http://www.naturalimagestatistics.net/nis_preprintFeb2009.pdf>. 14, 34, 129, 130

[14] BIOUCAS-DIAS, J. M., FIGUEIREDO, M. A. T., OLIVEIRA, J. P. "Adaptive Total Variation Image Deconvolution: A Majorization-Minimization Approach". In: *14th European Signal Processing Conference (EUSIPCO 2006)*, Florence, Italy, set. 2006. Disponível em: <http://www.eurasip.org/Proceedings/Eusipco/Eusipco2006/papers/1568982217.pdf>. 16, 22, 49

[15] BOVIK, A. *Handbook of Image and Video Processing*. New York, Academic Press, 2000. ISBN: 978-0121197926. 16

[16] HUANG, J. *Statistics of Natural Images and Models*. Tese de Doutorado, Brown University, Providence, Rhode Island, maio 2000. Disponível em: <http://www.dam.brown.edu/people/mumford/Papers/ThesesStudents/Huangthesis.pdf>. 16, 17, 18

[17] HUANG, J., MUMFORD, D. "Statistics of natural images and models". In: *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, v. 1, pp. 2 vol. (xxiii+637+663), 1999. doi:10.1109/CVPR.1999.786990. 16

[18] SRIVASTAVA, A., LEE, A., SIMONCELLI, E., et al. "On Advances in Statistical Modeling of Natural Images", *Journal of Mathematical Imaging and Vision*, v. 18, n. 1, pp. 17–33, 2003. doi:10.1023/A:1021889010444. 18

[19] SIMONCELLI, E., ADELSON, E. "Noise removal via Bayesian wavelet coring". In: *Image Processing, 1996. Proceedings., International Conference on*, v. 1, pp. 379–382, set. 1996. doi:10.1109/ICIP.1996.559512. 18

[20] RUDIN, L. I., OSHER, S., FATEMI, E. "Nonlinear total variation based noise removal algorithms", *Physica D.*, v. 60, pp. 259–268, 1992. Disponível em: <http://www.math-info.univ-paris5.fr/~lomn/Cours/ECE/PhysicaRudinOsher.pdf>. 19, 20, 22, 49, 52, 182

[21] CHOKSI, R., VAN GENNIP, Y., OBERMAN, A. "Anisotropic Total Variation Regularized L^1-Approximation and Denoising/Deblurring of 2D Bar Codes", *Arxiv preprint arXiv:1007.1035*, 2010. Disponível em: <http://www.math.mcgill.ca/rchoksi/pub/2dbarcode.pdf>. 20

[22] BIRKHOLZ, H. "A unifying approach to isotropic and anisotropic total variation denoising models", *Journal of Computational and Applied Mathematics*, v. 235, n. 8, pp. 2502–2514, fev. 2011. doi:10.1016/j.cam.2010.11.003. 20

[23] SETZER, S., STEIDL, G., TEUBER, T. "Restoration of images with rotated shapes", *Numerical Algorithms*, v. 48, n. 1, pp. 49–66, 2008. doi:10.1007/s11075-008-9182-y.

[24] ESEDOḠLU, S., OSHER, S. "Decomposition of images by the anisotropic Rudin-Osher-Fatemi model", *Communications on pure and applied mathematics*, v. 57, n. 12, pp. 1609–1626, 2004. Disponível em: <http://www.math.lsa.umich.edu/~esedoglu/Papers_Preprints/aniso_esedoglu_osher.pdf>. 20

[25] KAY, S. M. *Fundamentals of statistical signal processing: Estimation theory*. Upper Saddle River, NJ, USA, Prentice-Hall, Inc., 1993. ISBN: 0-13-345711-7. 22, 138, 144

[26] CHAMBOLLE, A., LIONS, P. L. "Image recovery via total variation minimization and related problems", *Numerische Mathematik*, v. 76, pp. 167–188, 1997. 22, 52

[27] CHAMBOLLE, A. "An Algorithm for Total Variation Minimization and Applications", *Journal of Mathematical Imaging and Vision*, v. 20, pp. 89–97, 2004. ISSN: 0924-9907. doi:10.1023/B:JMIV.0000011325.36760.1e. 53

[28] CHAMBOLLE, A. "Total Variation Minimization and a Class of Binary MRF Models". In: Rangarajan, A., Vemuri, B., Yuille, A. (Eds.), *Energy Minimization Methods in Computer Vision and Pattern Recognition*, v. 3757, *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 136–152, Augustine, FL, 2005. ISBN: 978-3-540-30287-2. doi:10.1007/11585978_10.

[29] CHAN, T., ESEDOGLU, S., PARK, F., et al. "Total Variation Image Restoration: Overview and Recent Developments". In: *Handbook of Mathematical Models in Computer Vision*, cap. 2, New York, Springer, 2005. ISBN: 978-0387263717. 52

[30] WOHLBERG, B., RODRÍGUEZ, P. "An Iteratively Reweighted Norm Algorithm for Minimization of Total Variation Functionals", *Signal Processing Letters, IEEE*, v. 14, n. 12, pp. 948–951, dez. 2007. ISSN: 1070-9908. doi:10.1109/LSP.2007.906221.

[31] AFONSO, M. V., BIOUCAS-DIAS, J. M., FIGUEIREDO, M. A. T. "Fast Image Recovery Using Variable Splitting and Constrained Optimization", *Image Processing, IEEE Transactions on*, v. 19, n. 9, pp. 2345–2356, set. 2010. ISSN: 1057-7149. doi:10.1109/TIP.2010.2047910. 22, 53, 71

[32] BRÁS, N. B., BIOUCAS-DIAS, J., MARTINS, R. C., et al. "An Alternating Direction Algorithm for Total Variation Reconstruction of Distributed Parameters", *Image Processing, IEEE Transactions on*, v. 21, n. 6, pp. 3004–3016, jun. 2012. ISSN: 1057-7149. doi:10.1109/TIP.2012.2188033. 22

[33] DANIELYAN, A., KATKOVNIK, V., EGIAZARIAN, K. "BM3D Frames and Variational Image Deblurring", *Image Processing, IEEE Transactions on*, v. 21, n. 4, pp. 1715–1728, abr. 2012. ISSN: 1057-7149. doi:10.1109/TIP.2011.2176954. 22, 136

[34] TAI, X.-C., WU, C. "Augmented Lagrangian Method, Dual Methods and Split Bregman Iteration for ROF Model". In: *Proceedings of the Second International Conference on Scale Space and Variational Methods in Computer Vision*, SSVM '09, pp. 502–513, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN: 978-3-642-02255-5. doi:10.1007/978-3-642-02256-2_42. 22

[35] CAI, J.-F., OSHER, S., SHEN, Z. "Split Bregman methods and frame based image restoration", *Multiscale modeling & simulation*, v. 8, n. 2, pp. 337–369, 2010. doi:10.1137/090753504. Disponível em: <http://www.math.nus.edu.sg/~matzuows/Image_Restoration.pdf>. 22, 23, 24

[36] BECK, A., TEBOULLE, M. "Fast Gradient-Based Algorithms for Constrained Total Variation Image Denoising and Deblurring Problems", *Image Processing, IEEE Transactions on*, v. 18, n. 11, pp. 2419–2434, nov. 2009. ISSN: 1057-7149. doi:10.1109/TIP.2009.2028250. 22

[37] BECK, A., TEBOULLE, M. "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems", *SIAM Journal on Imaging Sciences*, v. 2, n. 1, pp. 183–202, mar. 2009. ISSN: 1936-4954. doi:10.1137/080716542. Disponível em: <http://iew3.technion.ac.il/~becka/papers/71654.pdf>. 22

[38] OLIVEIRA, J. P., BIOUCAS-DIAS, J. M., FIGUEIREDO, M. A. "Adaptive total variation image deblurring: A majorization-minimization approach", *Signal Processing*, v. 89, n. 9, pp. 1683–1693, 2009. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0165168409001224>. 22, 36

[39] HESTENES, M. R. "Multiplier and gradient methods", *Journal of Optimization Theory and Applications*, v. 4, pp. 303–320, 1969. ISSN: 0022-3239. doi:10.1007/BF00927673. 22

[40] POWELL, M. J. D. "A method for nonlinear constraints in minimization problems". In: Fletcher, R. (Ed.), *Optimization*, Academic Press, pp. 283–298, New York, 1969. Disponível em: <http://www.ams.org/mathscinet-getitem?mr=42:7284>. 24, 68

[41] AFONSO, M. V., BIOUCAS-DIAS, J. M., FIGUEIREDO, M. A. T. "An Augmented Lagrangian Approach to the Constrained Optimization Formulation of Imaging Inverse Problems", *Image Processing, IEEE Transactions on*, v. 20, n. 3, pp. 681–695, mar. 2011. ISSN: 1057-7149. doi:10.1109/TIP.2010.2076294.

[42] CHAN, S. H., KHOSHABEH, R., GIBSON, K. B., et al. "An Augmented Lagrangian Method for Total Variation Video Restoration", *Image Processing, IEEE Transactions on*, v. 20, n. 11, pp. 3097–3111, nov. 2011. ISSN: 1057-7149. doi:10.1109/TIP.2011.2158229. 22, 23, 53, 54, 67, 68, 71

[43] NOCEDAL, J., WRIGHT, S. J. *Numerical Optimization*. 2nd ed. New York, Springer-Verlag, 2006. ISBN: 978-0387303031. 22, 23, 24, 53

[44] ECKSTEIN, J., BERTSEKAS, D. "On the Douglas—Rachford splitting method and the proximal point algorithm for maximal monotone operators", *Mathematical Programming*, v. 55, n. 1, pp. 293–318, 1992. 22, 68

[45] LI, C. *An Efficient Algorithm For Total Variation Regularization with Applications to the Single Pixel Camera and Compressive Sensing*. Tese de Mestrado, Rice University, set. 2009. 23, 54, 68

[46] YANG, J., ZHANG, Y., YIN, W. "An efficient TVL1 algorithm for deblurring multichannel images corrupted by impulsive noise", *SIAM Journal on Scientific Computing*, v. 31, n. 4, pp. 2842–2865, 2009.

[47] YANG, J., YIN, W., ZHANG, Y., et al. "A fast algorithm for edge-preserving variational multichannel image restoration", *SIAM Journal on Imaging Sciences*, v. 2, n. 2, pp. 569–592, 2009. 23

[48] TIBSHIRANI, R. "Regression shrinkage and selection via the LASSO", *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996. 23

[49] WANG, Y., YANG, J., YIN, W., et al. "A New Alternating Minimization Algorithm for Total Variation Image Reconstruction", *SIAM Journal on Imaging Sciences*, v. 1, n. 3, pp. 248–272, ago. 2008. 23

[50] BIOUCAS-DIAS, J., FIGUEIREDO, M. "An iterative algorithm for linear inverse problems with compound regularizers". In: *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pp. 685–688, San Diego, CA, out. 2008. doi:10.1109/ICIP.2008.4711847.

[51] HUANG, Y.-M., NG, M. K., WEN, Y.-W. "A Fast Total Variation Minimization Method for Image Restoration", *Multiscale Modeling & Simulation*, v. 7, n. 2, pp. 774–795, 2008.

[52] COMBETTES, P. L., PESQUET, J.-C. "Proximal Splitting Methods in Signal Processing". In: Bauschke, H. H., Burachik, R. S., Combettes, P. L., et al. (Eds.), *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, v. 49, *Springer Optimization and Its Applications*, Springer New York, cap. 10, pp. 185–212, New York, 2011. 33

[53] COMBETTES, P. L., PESQUET, J.-C. "Proximal Splitting Methods in Signal Processing", *ArXiv e-prints*, dez. 2009. Disponível em: <http://adsabs.harvard.edu/abs/2009arXiv0912.3522C>. 23

[54] BERTSEKAS, D. P. *Constrained Optimization and Lagrange Multiplier Methods*. Belmont, Massachusetts, Athena Scientific, 1996. Disponível em: <http://web.mit.edu/dimitrib/www/Constrained-Opt.pdf>. 24

[55] FORTIN, M., GLOWINSKI, R. *Augmented Lagrangian Methods*, v. 15. North-Holland, Amsterdam, Elsevier, 1983. 24

[56] DONG, W., ZHANG, L., SHI, G., et al. "Image Deblurring and Super-Resolution by Adaptive Sparse Domain Selection and Adaptive Regularization", *Image Processing, IEEE Transactions on*, v. 20, n. 7, pp. 1838–1857, jul. 2011. ISSN: 1057-7149. doi:10.1109/TIP.2011.2108306. 27, 35, 43, 44, 116, 123

[57] WU, X., DONG, W., ZHANG, X., et al. "Model-Assisted Adaptive Recovery of Compressed Sensing with Imaging Applications", *Image Processing, IEEE Transactions on*, v. 21, n. 2, pp. 451–458, fev. 2012. ISSN: 1057-7149. doi:10.1109/TIP.2011.2163520. 27, 34, 35, 43, 44, 116, 121, 123

[58] LI, X., ORCHARD, M. "New edge-directed interpolation", *Image Processing, IEEE Transactions on*, v. 10, n. 10, pp. 1521–1527, out. 2001. ISSN: 1057-7149. doi:10.1109/83.951537. 28, 80

[59] ZHANG, X., WU, X. "Image Interpolation by Adaptive 2-D Autoregressive Modeling and Soft-Decision Estimation", *Image Processing, IEEE Transactions on*, v. 17, n. 6, pp. 887–896, jun. 2008. ISSN: 1057-7149. doi:10.1109/TIP.2008.924279. 28, 35, 43, 44, 73, 79, 80, 87, 92, 104, 112, 123

[60] HUNG, K.-W., SIU, W.-C. "Robust Soft-Decision Interpolation Using Weighted Least Squares", *Image Processing, IEEE Transactions on*, v. 21, n. 3, pp. 1061 –1069, mar. 2012. ISSN: 1057-7149. doi:10.1109/TIP.2011.2168416. 28, 35, 80

[61] XIONG, R., DING, W., MA, S., et al. "A practical algorithm for tanner graph based image interpolation". In: *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pp. 1989–1992, Hong Kong, set. 2010. doi:10.1109/ICIP.2010.5648958. 28

[62] ZHANG, X., MA, S., ZHANG, Y., et al. "Nonlocal edge-directed interpolation". In: *10th Pacific Rim Conf. Multimedia—Advances in Multimedia Information Processing*, v. 5879, p. 1197–1207, 2009.

[63] LIU, X., ZHAO, D., XIONG, R., et al. "Image interpolation via regularized local linear regression". In: *PCS*, p. 118–121, dez. 2010.

[64] HUNG, K. W., SIU, W. C. "Improved image interpolation using bilateral filter for weighted least square estimation". In: *IEEE Int. Conf. Image Process.*, p. 3297–3300, Hong Kong, set. 2010.

[65] ZHANG, L., WU, X. "An edge-guided image interpolation algorithm via directional filtering and data fusion", *Image Processing, IEEE Transactions on*, v. 15, n. 8, pp. 2226–2238, ago. 2006. ISSN: 1057-7149. doi:10.1109/TIP.2006.877407. 28

[66] YU, G., SAPIRO, G., MALLAT, S. "Solving Inverse Problems With Piecewise Linear Estimators: From Gaussian Mixture Models to Structured Sparsity", *Image Processing, IEEE Transactions on*, v. 21, n. 5, pp. 2481–2499, maio 2012. ISSN: 1057-7149. doi:10.1109/TIP.2011.2176743. 28, 33, 34, 35, 45, 138, 139, 140, 141, 142

[67] PARK, S. C., PARK, M. K., KANG, M. G. "Super-resolution image reconstruction: a technical overview", *Signal Processing Magazine, IEEE*, v. 20, n. 3, pp. 21–36, maio 2003. ISSN: 1053-5888. doi:10.1109/MSP.2003.1203207. 28, 52, 174, 177, 178

[68] ELAD, M., MILANFAR, P., RUBINSTEIN, R. "Analysis versus synthesis in signal priors", *Inverse problems*, v. 23, pp. 947–968, 2007. 32, 44, 45, 126, 127, 128, 140, 141

[69] SELESNICK, I. W., FIGUEIREDO, M. A. T. "Signal restoration with overcomplete wavelet transforms: comparison of analysis and synthesis priors". In: *Proceedings of SPIE*, v. 7446, pp. 74460D–74460D–15, 2009. doi:10.1117/12.826663. Wavelets XIII. 45, 141

[70] ELAD, M. *Sparse and Redundant Representations*. New York, Springer, 2010. ISBN: 978-1-4419-7010-7. doi:10.1007/978-1-4419-7011-4. 127, 128, 129, 130, 141

[71] CANDÈS, E. J., ELDAR, Y. C., NEEDELL, D., et al. "Compressed sensing with coherent and redundant dictionaries", *Applied and Computational Harmonic Analysis*, v. 31, n. 1, pp. 59–73, 2011. ISSN: 1063-5203. doi:10.1016/j.acha.2010.10.002. 141

[72] OPHIR, B., ELAD, M., BERTIN, N., et al. "Sequential Minimal Eigenvalues-An Approach to Analysis Dictionary Learning". In: *19th European Signal Processing Conference (EUSIPCO 2011)*, pp. 1465–1469, set. 2011. 142

[73] YAGHOOBI, M., NAM, S., GRIBONVAL, R., et al. "Analysis operator learning for overcomplete cosparse representations". In: *The 19th European Signal Processing Conference (EUSIPCO-2011)*, pp. 1470–1474, set. 2011. 142

[74] ELAD, M. "The Analysis (Co-)Sparse Model". 2012. Disponível em: <http://www.cs.technion.ac.il/~elad/talks/2012/Analysis_Oberwolfach_2012.pdf>. 143

[75] NAM, S., DAVIES, M., ELAD, M., et al. "The cosparse analysis model and algorithms", *Applied and Computational Harmonic Analysis*, v. 0, n. 0, pp. –, 2012. ISSN: 1063-5203. doi:10.1016/j.acha.2012.03.006. to appear. 142

[76] CLEJU, N., JAFARI, M. G., PLUMBLEY, M. D. "Choosing Analysis or Synthesis Recovery for Sparse Reconstruction". In: *20th European Signal Processing Conference (EUSIPCO)*, pp. 869–873, Bucharest, Romania, ago. 2012. Disponível em: <http://www.eurasip.org/Proceedings/Eusipco/Eusipco2012/Conference/papers/1569581221.pdf>. 32, 45, 141

[77] ZIBULEVSKY, M., ELAD, M. "L1-L2 Optimization in Signal and Image Processing", *Signal Processing Magazine, IEEE*, v. 27, n. 3, pp. 76 –88, maio 2010. ISSN: 1053-5888. doi:10.1109/MSP.2010.936023. 32

[78] SLEIJPEN, G. L. G., FOKKEMA, D. R. "BiCGstab(ℓ) for linear equations involving unsymmetric matrices with complex spectrum", *Electronic Transactions on Numerical Analysis*, v. 1, pp. 11–32 (electronic only), set. 1993. ISSN: 1068-9613. 33, 86, 150

[79] MONTEFUSCO, L. B., LAZZARO, D. "An Iterative L1-Based Image Restoration Algorithm With an Adaptive Parameter Estimation", *Image Processing, IEEE Transactions on*, v. 21, n. 4, pp. 1676–1686, abr. 2012. ISSN: 1057-7149. doi:10.1109/TIP.2011.2173205. 33, 36

[80] ROTH, S., BLACK, M. J. "Fields of experts", *International Journal of Computer Vision*, v. 82, n. 2, pp. 205–229, 2009. 33, 132, 133, 134, 135

[81] DABOV, K., FOI, A., KATKOVNIK, V., et al. "Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering", *Image Processing, IEEE Transactions on*, v. 16, n. 8, pp. 2080–2095, ago. 2007. ISSN: 1057-7149. doi:10.1109/TIP.2007.901238. 33, 45, 136, 137

[82] MCLACHLAN, G. J., KRISHNAN, T. *The EM Algorithm and Extensions*. Wiley Series in Probability and Statistics. 2 ed. Hoboken, New Jersey, Wiley-Interscience, 2008. ISBN: 978-0-471-20170-0. 35

[83] GUO, K., YANG, X., ZHA, H., et al. "Multiscale Semilocal Interpolation With Antialiasing", *Image Processing, IEEE Transactions on*, v. 21, n. 2, pp. 615–625, fev. 2012. ISSN: 1057-7149. doi:10.1109/TIP.2011.2165290. 35

[84] PROTTER, M., ELAD, M., TAKEDA, H., et al. "Generalizing the Nonlocal-Means to Super-Resolution Reconstruction", *Image Processing, IEEE Transactions on*, v. 18, n. 1, pp. 36–51, jan. 2009. ISSN: 1057-7149. doi:10.1109/TIP.2008.2008067. 35, 43, 105, 108, 109, 119, 123, 135, 180, 181

[85] TAKEDA, H., MILANFAR, P., PROTTER, M., et al. "Super-Resolution Without Explicit Subpixel Motion Estimation", *Image Processing, IEEE Transactions on*, v. 18, n. 9, pp. 1958–1975, set. 2009. ISSN: 1057-7149. doi:10.1109/TIP.2009.2023703. 35, 43, 105, 106, 180, 182, 184

[86] BAUER, F., LUKAS, M. A. "Comparing parameter choice methods for regularization of ill-posed problems", *Mathematics and Computers in Simulation*, v. 81, n. 9, pp. 1795–1841, 2011. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0378475411000607>. 36

[87] YUAN, Q., ZHANG, L., SHEN, H., et al. "Adaptive Multiple-Frame Image Super-Resolution Based on U-Curve", *Image Processing, IEEE Transactions on*, v. 19, n. 12, pp. 3157–3170, dez. 2010. ISSN: 1057-7149. doi:10.1109/TIP.2010.2055571. 36

[88] ZIBETTI, M. V., BAZÁN, F. S., MAYER, J. "Determining the regularization parameters for super-resolution problems", *Signal Processing*, v. 88, n. 12, pp. 2890 – 2901, 2008. ISSN: 0165-1684. doi:10.1016/j.sigpro.2008.06.010. 36

[89] ZIBETTI, M. V., BAZÁN, F. S., MAYER, J. "Estimation of the parameters in regularized simultaneous super-resolution", *Pattern Recognition Letters*, v. 32, n. 1, pp. 69 – 78, 2011. ISSN: 0167-8655. doi:10.1016/j.patrec.2009.12.009. 36

[90] ZIBETTI, M. V. W., MAYER, J. "A Robust and Computationally Efficient Simultaneous Super-Resolution Scheme for Image Sequences", *Circuits and Systems for Video Technology, IEEE Transactions on*, v. 17, n. 10, pp. 1288–1300, out. 2007. ISSN: 1051-8215. doi:10.1109/TCSVT.2007.903801. 36, 37, 45

[91] RATLIFF, B. M. *A Generalized Algebraic Scene-based Nonuniformity Correction Algorithm*. Tese de Doutorado, The University of New Mexico, 2004. 39, 40, 145, 147, 148, 149, 159, 162, 166

[92] SCRIBNER, D. A., KRUER, M. R., GRIDLEY, J. C., et al. "Physical Limitations to Nonuniformity Correction in IR Focal Plane Arrays", *Proceedings of SPIE*, v. 865, pp. 185–202, maio 1988. doi:10.1117/12.943565. 40

[93] PERRY, D. L., DERENIAK, E. L. "Linear Theory of Nonuniformity Correction in Infrared Staring Sensors", *Optical Engineering*, v. 32, n. 8, pp. 1853–1859, 1993. 40, 145, 147

[94] PIPA, D. R., DA SILVA, E. A. B., PAGLIARI, C. L. "Correção de não uniformidade em vídeo infravermelho por gradiente descendente". In: *XXVI Simpósio Brasileiro de Telecomunicações - SBrT'08*, 2008. 41, 48

[95] AHARON, M., ELAD, M., BRUCKSTEIN, A. "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation", *Signal Processing, IEEE Transactions on*, v. 54, n. 11, pp. 4311–4322, nov. 2006. ISSN: 1053-587X. doi:10.1109/TSP.2006.881199. 44, 130, 131

[96] CHATTERJEE, P., MILANFAR, P. "Patch-Based Near-Optimal Image Denoising", *Image Processing, IEEE Transactions on*, v. 21, n. 4, pp. 1635–1649, abr. 2012. ISSN: 1057-7149. doi:10.1109/TIP.2011.2172799. 45, 138

[97] PIPA, D. R., DA SILVA, E. A. B., PAGLIARI, C. L., et al. "Joint Bias and Gain Nonuniformity Correction of Infrared Videos Using Tensorial-RLS Technique". In: *Image Processing, 2009. ICIP 2009. 16th IEEE International Conference on*, 2009. 46, 48, 147, 158

[98] OKAMOTO, J., GRASSI, V., AMARAL, P., et al. "Development of an Autonomous Robot for Gas Storage Spheres Inspection", *Journal of Intelligent and Robotic Systems*, v. 66, pp. 23–35, 2012. ISSN: 0921-0296. doi:10.1007/s10846-011-9607-z. 47

[99] PIPA, D., MORIKAWA, S., PIRES, G., et al. "Flexible Riser Monitoring Using Hybrid Magnetic/Optical Strain Gage Techniques through RLS Adaptive Filtering", *EURASIP Journal on Advances in Signal Processing*, v. 2010, n. 1, pp. 176–203, 2010. ISSN: 1687-6180. doi:10.1155/2010/176203. 47

[100] OKAMOTO, J., GRASSI, V., AMARAL, P. F. S., et al. "Autonomous robot for welding line following for gas storage spheres inspection". In: *Mechatronics2010 - The 12th Mechatronics Forum Biennial International Conference*, Zurich, 2010. 48

[101] MORIKAWA, S. R. K., CAMERINI, C. S., PIPA, D. R., et al. "Monitoring of Flexible Oil Lines Using FBG Sensors". In: *International Conference on Optical Fibre Sensors (OFS)*, Perth, Australia, 2008. 48

[102] MARINHO, M. G., CAMERINI, C. S., MORIKAWA, S. R. K., et al. "NEW TECHNIQUES FOR INTEGRITY MANAGEMENT OF FLEXIBLE RISER-END FITTING CONNECTION". In: *27th International Conference on Offshore Mechanics and Arctic Engineering (OMAE2008)*, Estoril, Portugal, 2008. 48

[103] PIPA, D. R. "DENOISING DE SINAIS DE ULTRA-SOM UTILIZANDO WAVELETS". In: *XXVI Congresso Nacional de Ensaios Não Destrutivos e Inspeção (CONAEND&IEV)*, São Paulo, 2008. 48

[104] PIPA, D. R., PIRES, G. P. "MONITORAMENTO DA TORÇÃO EM RISERS FLEXÍVEIS ATRAVÉS DE PROCESSAMENTO DE IMAGENS". In: *XXVI Congresso Nacional de Ensaios Não Destrutivos e Inspeção (CONAEND&IEV)*, São Paulo, 2008. 48

[105] PIRES, G. P., PIPA, D. R. "CLASSIFICAÇÃO DE RUÍDOS EM EMISSÃO ACÚSTICA UTILIZANDO REDES NEURAIS". In: *XXVI Congresso Nacional de Ensaios Não Destrutivos e Inspeção (CONAEND&IEV)*, São Paulo, 2008. 48

[106] CHAN, T. F., ESEDOGLU, S., PARK, F. E. "Image decomposition combining staircase reduction and texture extraction", *Journal of Visual Communication and Image Representation*, v. 18, n. 6, pp. 464–486, dez. 2007. 52

[107] STEFAN, W., RENAUT, R., GELB, A. "Improved total variation-type regularization using higher order edge detectors", *SIAM Journal on Imaging Sciences*, pp. 232–251, jan. 2010. 52

[108] FARSIU, S., ROBINSON, M. D., ELAD, M., et al. "Fast and robust multiframe super resolution", *Image Processing, IEEE Transactions on*, v. 13, n. 10, pp. 1327–1344, out. 2004. ISSN: 1057-7149. doi:10.1109/TIP.2004.834669. 52, 105, 179, 182, 192

[109] KIRIYAMA, S., USUI, T., GOTO, T., et al. "Diagonal Total Variation Regularization Criterion for Fast Convergence". In: *Proc. ICALIP*, pp. 1494–1498, nov. 2010. 53

[110] KARAHANOĞLU, F. I., İLKER BAYRAM, VILLE, D. V. D. "A Signal Processing Approach to Generalized 1-D Total Variation", *Signal Processing, IEEE Transactions on*, v. 59, n. 11, pp. 5265–5274, nov. 2011. ISSN: 1053-587X. doi:10.1109/TSP.2011.2164399. 53

[111] DINIZ, P. S. R., DA SILVA, E. A. B., NETTO, S. L. *Digital Signal Processing: System Analysis and Design*. 2nd ed. Cambridge, Cambridge University Press, 2010. 57

[112] ROCKAFELLAR, R. T., WETS, R. J. B. "Subderivatives and Subgradients". In: *Variational Analysis*, v. 317, *Grundlehren der mathematischen Wissenschaften*, Springer Berlin Heidelberg, pp. 298–348, Dordrecht, 1998. ISBN: 978-3-642-02431-3. 68

[113] VAN DER VORST, H. A. "BI-CGSTAB: a fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems", *SIAM Journal on Scientific and Statistical Computing*, v. 13, n. 2, pp. 631–644, 1992. ISSN: 0196-5204. doi:10.1137/0913035. 86

[114] MALLAT, S., YU, G. "Structured pursuits for geometric super-resolution". In: *16th IEEE International Conference on Image Processing (ICIP)*, pp. 1477–1480, Cairo, Egypt, nov. 2009. 87, 104

[115] ELAD, M., HEL-OR, Y. "A fast super-resolution reconstruction algorithm for pure translational motion and common space-invariant blur", *Image Processing, IEEE Transactions on*, v. 10, n. 8, pp. 1187–1193, ago. 2001. ISSN: 1057-7149. doi:10.1109/83.935034. 105, 177, 182

[116] GRAY, R. M. *Toeplitz and Circulant Matrices: A Review*. Norwell, Massachusetts, Now Publishers, 2006. 105, 179

[117] HORN, B. K. P., SCHUNCK, B. G. "Determining Optical Flow", *Artificial Intelligence*, v. 17, pp. 185–203, 1981. 107

[118] LUCAS, B. D., KANADE, T. "An Iterative Image Registration Technique with an Application to Stereo Vision". In: *7th International Joint Conference on Artificial Intelligence - IJCAI'81*, v. 2, pp. 674–679, abr. 1981. 107, 162

[119] BOUGUET, J.-Y. *Pyramidal Implementation of the Lucas-Kanade Feature Tracker: Description of the algorithm*. Relatório técnico, Intel Microprocessor Research Labs, 2000. OpenCV Document. 107

[120] LI, X. "Video Processing Via Implicit and Mixture Motion Models", *Circuits and Systems for Video Technology, IEEE Transactions on*, v. 17, n. 8,

pp. 953–963, ago. 2007. ISSN: 1051-8215. doi:10.1109/TCSVT.2007.896656. 108

[121] WU, X., ZHANG, X., WANG, J. "Model-guided adaptive recovery of compressive sensing". In: *Data Compression Conference, 2009. DCC'09.*, pp. 123–132. IEEE, 2009. 116

[122] BUADES, A., COLL, B., MOREL, J. M. "A Review of Image Denoising Algorithms, with a New One", *Multiscale Modeling & Simulation*, v. 4, n. 2, pp. 490–530, 2005. doi:10.1137/040616024. Disponível em: <http://hal.archives-ouvertes.fr/docs/00/27/11/41/PDF/061602r.pdf>. 123, 135, 138

[123] CHEN, S. S., DONOHO, D. L., SAUNDERS, M. A. "Atomic Decomposition by Basis Pursuit", *SIAM Journal on Scientific Computing*, v. 20, n. 1, pp. 33–61, 1998. ISSN: 1095-7197. doi:10.1137/S1064827596304010. Disponível em: <http://www.stanford.edu/group/SOL/papers/BasisPursuit-SIGEST.pdf>. 126

[124] ELAD, M., STARCK, J. L., QUERRE, P., et al. "Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA)", *Applied Computational Harmonic Analysis*, v. 19, n. 3, pp. 340–358, nov. 2005. 128

[125] FOI, A., DABOV, K., KATKOVNIK, V., et al. "Shape-adaptive DCT for denoising and image reconstruction". In: *Proceedings of SPIE*, v. 6064, pp. 203–214, 2006.

[126] GULERYUZ, O. "Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising-part I: theory", *Image Processing, IEEE Transactions on*, v. 15, n. 3, pp. 539–554, mar. 2006. ISSN: 1057-7149. doi:10.1109/TIP.2005.863057.

[127] GULERYUZ, O. "Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising-part II: adaptive algorithms", *Image Processing, IEEE Transactions on*, v. 15, n. 3, pp. 555–571, mar. 2006. ISSN: 1057-7149. doi:10.1109/TIP.2005.863055.

[128] FADILI, M. J., STARCK, J. L., MURTAGH, F. "Inpainting and zooming using sparse representations", *The Computer Journal*, v. 52, n. 1, pp. 64–79, jan. 2009. doi:10.1093/comjnl/bxm055.

[129] SADEGHIPOUR, Z., BABAIE-ZADEH, M., JUTTEN, C. "An adaptive thresholding approach for image denoising using redundant representations". In: *Machine Learning for Signal Processing, 2009. MLSP 2009. IEEE International Workshop on*, pp. 1 –6, set. 2009. doi: 10.1109/MLSP.2009.5306213. 128

[130] MALLAT, S. *A Wavelet Tour of Signal Processing*. 3rd ed. Burlington, MA, Academic Press, 2009. ISBN: 978-0-12-374370-1. 128, 172

[131] CANDÈS, E. J., DONOHO, D. L. *Curvelets*. Relatório técnico, Standford, 1999. Disponível em: <http://www-stat.stanford.edu/~donoho/Reports/1999/curvelets.pdf>. 128

[132] CANDÈS, E. J., DONOHO, D. L. "Curvelets - A surprisingly effective nonadaptive representation for objects with edges". In: Cohen, A., Rabut, C., Schumaker, L. L. (Eds.), *Curve and Surface Fitting: Saint-Malo*, pp. 105–120, Nashville, Tennessee, 1999. Vanderbilt University Press. Disponível em: <http://www-stat.stanford.edu/~candes/papers/Curvelet-SMStyle.ps.gz>.

[133] STARCK, J.-L., CANDES, E., DONOHO, D. "The curvelet transform for image denoising", *Image Processing, IEEE Transactions on*, v. 11, n. 6, pp. 670–684, jun. 2002. ISSN: 1057-7149. doi:10.1109/TIP.2002.1014998. 128

[134] CANDÈS, E. J. "Harmonic Analysis of Neural Networks", *Applied and Computational Harmonic Analysis*, v. 6, n. 2, pp. 197–218, mar. 1999. ISSN: 1063-5203. doi:10.1006/acha.1998.0248. Disponível em: <http://www-stat.stanford.edu/~emmanuel/papers/net.ps>. 128

[135] CANDÈS, E. J. "Ridgelets and their Derivatives: Representation of Images with Edges". In: Cohen, A., Rabut, C., Schumaker, L. L. (Eds.), *Curve and Surface Fitting: Saint-Malo*, pp. 95–104, Nashville, Tennessee, 1999. Vanderbilt University Press. Disponível em: <http://www-stat.stanford.edu/~candes/papers/StMalo.ps.gz>. 128

[136] DO, M. N., VETTERLI, M. "The contourlet transform: an efficient directional multiresolution image representation", *Image Processing, IEEE Transactions on*, v. 14, n. 12, pp. 2091–2106, dez. 2005. ISSN: 1057-7149. doi:10.1109/TIP.2005.859376. 128

[137] DONOHO, D. L. "For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution", *Communica-*

tions on Pure and Applied Mathematics, v. 59, n. 6, pp. 797–829, 2006. ISSN: 1097-0312. doi:10.1002/cpa.20132. 129

[138] ELAD, M., AHARON, M. "Image Denoising Via Sparse and Redundant Representations Over Learned Dictionaries", *Image Processing, IEEE Transactions on*, v. 15, n. 12, pp. 3736–3745, dez. 2006. ISSN: 1057-7149. doi:10.1109/TIP.2006.881969. 129

[139] BIOUCAS-DIAS, J. "Bayesian wavelet-based image deconvolution: a GEM algorithm exploiting a class of heavy-tailed priors", *Image Processing, IEEE Transactions on*, v. 15, n. 4, pp. 937–951, abr. 2006. ISSN: 1057-7149. doi:10.1109/TIP.2005.863972. 138

[140] FIGUEIREDO, M., BIOUCAS-DIAS, J., NOWAK, R. "Majorization-Minimization Algorithms for Wavelet-Based Image Restoration", *Image Processing, IEEE Transactions on*, v. 16, n. 12, pp. 2980–2991, dez. 2007. ISSN: 1057-7149. doi:10.1109/TIP.2007.909318.

[141] ANBARJAFARI, G., DEMIREL, H. "Image Super Resolution Based on Interpolation of Wavelet Domain High Frequency Subbands and the Spatial Domain Input Image", *ETRI Journal*, v. 32, n. 3, jun. 2010. 129

[142] STARCK, J.-L., ELAD, M., DONOHO, D. "Image decomposition via the combination of sparse representations and a variational approach", *Image Processing, IEEE Transactions on*, v. 14, n. 10, pp. 1570–1582, out. 2005. ISSN: 1057-7149. doi:10.1109/TIP.2005.852206. 129

[143] OLSHAUSEN, B. A., FIELD, D. J. "Emergence of simple-cell receptive field properties by learning a sparse code for natural images", *Nature*, v. 381, n. 13, jun. 1996. 129

[144] BELL, A. J., SEJNOWSKI, T. J. "The independent components of natural scenes are edge filters", *Vision Research*, v. 37, n. 23, pp. 3327–3338, 1997. ISSN: 0042-6989. doi:10.1016/S0042-6989(97)00121-1. 129

[145] HYVÄRINEN, A., KARHUNEN, J., OJA, E. *Independent component analysis*. New York, NY, USA, Wiley Interscience, 2001. 129

[146] HYVÄRINEN, A., HOYER, P., INKI, M. "Topographic ICA as a model of V1 receptive fields". In: *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, v. 4, pp. 83–88, 2000. doi:10.1109/IJCNN.2000.860754. 130

[147] HYVÄRINEN, A., HOYER, P., INKI, M. "Topographic ICA as a Model of Natural Image Statistics". In: Lee, S.-W., Bülthoff, H., Poggio, T. (Eds.), *Biologically Motivated Computer Vision*, v. 1811, *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 447–469, Berlin, 2000. ISBN: 978-3-540-67560-0. doi:10.1007/3-540-45482-9_54. 130

[148] HYVÄRINEN, A., HOYER, P. "Emergence of complex cell properties by decomposition of natural images into independent feature subspaces". In: *Artificial Neural Networks, 1999. ICANN 99. Ninth International Conference on (Conf. Publ. No. 470)*, v. 1, pp. 257 –262 vol.1, 1999. doi:10.1049/cp:19991118. 130

[149] HYVÄRINEN, A., HOYER, P. "Emergence of Phase- and Shift-Invariant Features by Decomposition of Natural Images into Independent Feature Subspaces", *Neural Computation*, v. 12, n. 7, pp. 1705–1720, jul. 2000. doi:10.1162/089976600300015312.

[150] HYVÄRINEN, A., KÖSTER, U. "Complex cell pooling and the statistics of natural images", *Network: Computation in Neural Systems*, v. 18, n. 2, pp. 81–100, 2007. 130

[151] GEMAN, S., GEMAN, D. "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images", *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. PAMI-6, n. 6, pp. 721–741, nov. 1984. ISSN: 0162-8828. doi:10.1109/TPAMI.1984.4767596. 131, 132

[152] PÉREZ, P. "Markov random fields and images", *CWI Quarterly*, v. 11, n. 4, pp. 413–437, 1998. Disponível em: <http://www-sigproc.eng.cam.ac.uk/research/reading%20group/material/perez.pdf>. 131

[153] LI, S. Z. *Markov Random Field Modeling in Image Analysis*. Advances in Pattern Recognition. 3 ed. London, Springer-Verlag, 2009. ISBN: 978-1-84800-278-4. doi:10.1007/978-1-84800-279-1. 132, 133

[154] GEMAN, D., REYNOLDS, G. "Constrained restoration and the recovery of discontinuities", *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 14, n. 3, pp. 367–383, mar. 1992. ISSN: 0162-8828. doi:10.1109/34.120331. 133

[155] BOUMAN, C., SAUER, K. "A generalized Gaussian image model for edge-preserving MAP estimation", *Image Processing, IEEE Transactions on*, v. 2, n. 3, pp. 296–310, jul. 1993. ISSN: 1057-7149. doi:10.1109/83.236536. 133

[156] CHANTAS, G. K., GALATSANOS, N. P., LIKAS, A. C. "Bayesian Restoration Using a New Nonstationary Edge-Preserving Image Prior", *Image Processing, IEEE Transactions on*, v. 15, n. 10, pp. 2987–2997, out. 2006. ISSN: 1057-7149. doi:10.1109/TIP.2006.877520.

[157] BABACAN, S., MOLINA, R., KATSAGGELOS, A. "Generalized Gaussian Markov random field image restoration using variational distribution approximation". In: *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pp. 1265 –1268, abr. 2008. doi:10.1109/ICASSP.2008.4517847.

[158] CHEN, J., NUNEZ-YANEZ, J., ACHIM, A. "Video super-resolution using generalized Gaussian Markov Random Fields", *IEEE Signal Processing Letters*, v. 19, n. 2, fev. 2012. 133

[159] ZHU, S. C., MUMFORD, D. "Prior learning and Gibbs reaction-diffusion", *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 19, n. 11, pp. 1236–1250, nov. 1997. ISSN: 0162-8828. doi:10.1109/34.632983. 133, 135

[160] HINTON, G. "Products of experts". In: *Int'l Conf. on Artificial Neural Networks*, v. 1, p. 1–6, Edinburgh, GB, 1999.

[161] TEH, Y. W., WELLING, M., HINTON, S. O. G. E. "Energy-based models for sparse overcomplete representations", *The Journal of Machine Learning Research*, v. 4, pp. 1235–1260, 2003. Disponível em: <http://jmlr.csail.mit.edu/papers/volume4/teh03a/teh03a.pdf>. 133

[162] WELLING, M., HINTON, G. E., OSINDERO, S. "Learning sparse topographic representations with products of Student-t distributions", *Advances in Neural Information Processing Systems (NIPS)*, v. 15, pp. 1359–1366, 2003. Disponível em: <http://books.nips.cc/papers/files/nips15/VS15.pdf>. 133

[163] CHARBONNIER, P., BLANC-FERAUD, L., AUBERT, G., et al. "Deterministic edge-preserving regularization in computed imaging", *Image Processing, IEEE Transactions on*, v. 6, n. 2, pp. 298–311, fev. 1997. ISSN: 1057-7149. doi:10.1109/83.551699. 133

[164] WEISS, Y., FREEMAN, W. T. "What makes a good model of natural images?" In: *CVPR 2007: Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society*, pp. 1–8, 2007. 134

[165] LECUN, Y., HUANG, F. J. "Loss Functions for Discriminative Training of Energy-Based Models". In: *10th International Workshop on Artificial Intelligence and Statistics (AIStats'05*, 2005. 134

[166] WILLIAMS, C. K. I., AGAKOV, F. V. "Products of Gaussians and probabilistic minor component analysis", *Neural Computational*, v. 14, n. 5, pp. 1169–1182, maio 2002. ISSN: 0899-7667. doi:10.1162/089976602753633439. 134

[167] KATKOVNIK, V., FOI, A., EGIAZARIAN, K., et al. "From Local Kernel to Nonlocal Multiple-Model Image Denoising", *International Journal of Computer Vision*, v. 86, pp. 1–32, 2010. ISSN: 0920-5691. doi:10.1007/s11263-009-0272-7. 135

[168] AWATE, S. P., WHITAKER, R. T. "Unsupervised, information-theoretic, adaptive image filtering for image restoration", *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 28, n. 3, pp. 364–376, mar. 2006. ISSN: 0162-8828. doi:10.1109/TPAMI.2006.64. 135, 138

[169] KERVRANN, C., BOULANGER, J. "Optimal Spatial Adaptation for Patch-Based Image Denoising", *Image Processing, IEEE Transactions on*, v. 15, n. 10, pp. 2866–2878, out. 2006. ISSN: 1057-7149. doi:10.1109/TIP.2006.877529. 135

[170] CHATTERJEE, P., MILANFAR, P. "Practical Bounds on Image Denoising: From Estimation to Information", *Image Processing, IEEE Transactions on*, v. 20, n. 5, pp. 1221–1233, maio 2011. ISSN: 1057-7149. doi:10.1109/TIP.2010.2092440. 138

[171] CHATTERJEE, P. *Patch-based Image Denoising and Its Performance Limits*. Tese de Doutorado, University of California, Santa Cruz, jun. 2011. 138

[172] PORTILLA, J., STRELA, V., WAINWRIGHT, M. J., et al. "Image denoising using scale mixtures of Gaussians in the wavelet domain", *Image Processing, IEEE Transactions on*, v. 12, n. 11, pp. 1338–1351, nov. 2003. ISSN: 1057-7149. doi:10.1109/TIP.2003.818640. 138

[173] OSINDERO, S., WELLING, M., HINTON, G. E. "Topographic product models applied to natural scene statistics", *Neural Computation*, v. 18, n. 2, pp. 381–414, fev. 2006. doi:10.1162/089976606775093936. Disponível em: <http://www.ics.uci.edu/~welling/publications/papers/PoT_NC2005_AcceptedDraft.pdf>.

[174] GUERRERO-COLON, J., MANCERA, L., PORTILLA, J. "Image Restoration Using Space-Variant Gaussian Scale Mixtures in Overcomplete Pyramids", *Image Processing, IEEE Transactions on*, v. 17, n. 1, pp. 27–41, jan. 2008. ISSN: 1057-7149. doi:10.1109/TIP.2007.911473. 138

[175] FIGUEIREDO, M., JAIN, A. "Unsupervised learning of finite mixture models", *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 24, n. 3, pp. 381–396, mar. 2002. ISSN: 0162-8828. doi:10.1109/34.990138. 139

[176] CHOUDREY, R. A. *Variational Methods for Bayesian Independent Component Analysis*. Tese de Doutorado, University of Oxford, 2002. Disponível em: <http://www.robots.ox.ac.uk/~parg/pubs/theses/RizwanChoudrey_thesis.pdf>. 139

[177] BILMES, J. A. *A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models*. Relatório técnico, International Computer Science Institute, Berkley, CA, 1998. Disponível em: <http://www.icsi.berkeley.edu/ftp/global/pub/techreports/1997/tr-97-021.pdf>. 140

[178] PORTILLA, J. "Image restoration through l0 analysis-based sparse optimization in tight frames". In: *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pp. 3909–3912, nov. 2009. doi:10.1109/ICIP.2009.5413975. 141, 142

[179] GIRYES, R., ELAD, M. "CoSaMP and Subspace-Pursuit (SP) for the Cosparse analysis model". In: *20th European Signal Processing Conference (EUSIPCO)*, pp. 964–968, Bucharest, Romania, ago. 2012. Disponível em: <http://www.eurasip.org/Proceedings/Eusipco/Eusipco2012/Conference/papers/1569577749.pdf>. 141

[180] MALLAT, S., YU, G. "Super-Resolution With Sparse Mixing Estimators", *Image Processing, IEEE Transactions on*, v. 19, n. 11, pp. 2889–2900, nov. 2010. ISSN: 1057-7149. doi:10.1109/TIP.2010.2049927. 141

[181] RUBINSTEIN, R., FAKTOR, T., ELAD, M. "K-SVD dictionary-learning for the analysis sparse model". In: *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 5405 –5408, mar. 2012. doi:10.1109/ICASSP.2012.6289143. 142

[182] YAGHOOBI, M., NAM, S., GRIBONVAL, R., et al. "Noise aware analysis operator learning for approximately cosparse signals". In: *Acoustics, Speech*

and Signal Processing (ICASSP), 2012 IEEE International Conference on, pp. 5409 –5412, mar. 2012. doi:10.1109/ICASSP.2012.6289144.

[183] NAM, S., DAVIES, M., ELAD, M., et al. "Cosparse analysis modeling - uniqueness and algorithms". In: *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 5804 –5807, maio 2011. doi:10.1109/ICASSP.2011.5947680.

[184] CHIERCHIA, G., PUSTELNIK, N., PESQUET, J., et al. "A proximal approach for constrained cosparse modelling". In: *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 3433 –3436, mar. 2012. doi:10.1109/ICASSP.2012.6288654. 142

[185] PEEBLES, P. Z. *Probability, Random Variables, and Random Signal Principles*. New York, McGraw-Hill Inc., 1993. ISBN: 0-07-366007-8. 143

[186] WASSERMAN, L. *All of Statistics: A Concise Course in Statistical Inference*. Springer Texts in Statistics. New York, Springer, 2004. ISBN: 978-0-387-40272-7. 144

[187] MILTON, A. F., BARONE, F. R., KRUER, M. R. "Influence of nonuniformity on infrared focal plane array performance", *Optical Engineering*, v. 24, pp. 855–862, 1985. 145

[188] HARDIE, R. C., DROEGE, D. R. "A MAP estimator for simultaneous superresolution and detector nonuniformity correction", *EURASIP Journal on Advances in Signal Processing*, v. 2007, n. 1, pp. 206–206, 2007. ISSN: 1110-8657. doi:10.1155/2007/89354. 146, 147, 148

[189] TORRES, S. N., HAYAT, M. M. "Kalman Filtering for Adaptive Nonuniformity Correction in Infrared Focal Plane Arrays", *Journal of the Optical Society of America*, v. 20, n. 3, pp. 470–480, 2003. 148

[190] TORRES, S. N., PEZOA, J. E., HAYAT, M. M. "Scene-based Nonuniformity Correction for Focal Plane Arrays Using the Method of the Inverse Covariance Form", *Applied Optics*, v. 42, n. 29, pp. 5872–5881, Oct. 2003. 148, 149

[191] HARDIE, R. C., HAYAT, M. M., ARMSTRONG, E. E., et al. "Scene-based Nonuniformity Correction with Video Sequences and Registration", *Applied Optics*, v. 39, n. 8, pp. 1241–1250, 2000. 148, 159

[192] AVERBUCH, A., LIRON, G., BOBROVSKY, B. Z. "Scene based non-uniformity correction in thermal images using Kalman filter", *Image*

and Vision Computing, v. 25, n. 6, pp. 833–851, 2007. ISSN: 0262-8856. doi:10.1016/j.imavis.2006.05.019. 148, 158, 167

[193] SAKOĞLU, U., HARDIE, R. C., HAYAT, M. M., et al. "An algebraic restoration method for estimating fixed pattern noise in infrared imagery from a video sequence". In: *49th Annual Meeting SPIE: Applications of Digital Image Processing XXVII*, v. 5558, pp. 69–79, Denver, CO, ago. 2004. SPIE Proceeding. 147, 148

[194] HARRIS, J. G., CHIANG, Y.-M. "An Analog Implementation of the Constant Average Statistics Constraint For Sensor Calibration". In: Mozer, M. C., Jordan, M. I., Petsche, T. (Eds.), *Proc. Advances in Neural Information Processing Systems*, v. 9, p. 699. The MIT Press, 1997. Disponível em: <citeseer.ist.psu.edu/article/harris97analog.html>. 148

[195] HAYAT, M. M., TORRES, S. N., ARMSTRONG, E., et al. "Statistical Algorithm for Nonuniformity Correction in Focal-Plane Arrays", *Applied Optics*, v. 38, pp. 772–780, 1990. 148

[196] SCRIBNER, D., SARKADY, K., KRUER, M., et al. "Adaptive retina-like preprocessing for imaging detector arrays", *Neural Networks, 1993., IEEE International Conference on*, pp. 1955–1960 vol.3, 1993. doi:10.1109/ICNN.1993.298856. 148

[197] RATLIFF, B. M., HAYAT, M. M., HARDIE, R. C. "An Algebraic Algorithm for Nonuniformity Correction in Focal Plane Arrays", *Journal of the Optical Society of America*, v. 19, n. 9, pp. 1737–1747, 2002. 148

[198] TORRES, F., MARTIN, C., TORRES, S. "A RLS Filter for Nonuniformity and Ghosting Correction of Infrared Image Sequences". In: Martínez-Trinidad, J., Carrasco Ochoa, J., Kittler, J. (Eds.), *Progress in Pattern Recognition, Image Analysis and Applications*, v. 4225, *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 446–454, Berlin, 2006. ISBN: 978-3-540-46556-0. doi:10.1007/11892755_46. 148

[199] DINIZ, P. S. R. *Adaptive filtering: Algorithms and Practical Implementations (3rd ed.)*. Boston, MA, Springer, 2008. 150, 151, 152, 155, 156

[200] HAYKIN, S. *Adaptive filter theory (3rd ed.)*. Upper Saddle River, NJ, USA, Prentice-Hall, Inc., 1996. ISBN: 0-13-322760-X. 150, 151

[201] MADISETTI, V. K., WILLIAMS, D. B. *The Digital Signal Processing Handbook*. Boca Raton, Florida, CRC Press LLC, 1999. 150

[202] DATTORRO, J. *Convex Optimization and Euclidean Distance Geometry*. Palo Alto, Meboo, 2008. 150, 152

[203] WERNER, S., DINIZ, P. S. R. "Set-Membership Affine Projection Algorithm", *IEEE Signal Processing Letters*, v. 8, pp. 231–235, ago. 2001. 155

[204] SANKARAN, S. G., BEEX, A. A. L. "Convergence behavior of affine projection algorithms", *Signal Processing, IEEE Transactions on*, v. 48, n. 4, pp. 1086–1096, abr. 2000. ISSN: 1053-587X. doi:10.1109/78.827542. 155

[205] SONI, R. A., GALLIVAN, K. A., JENKINS, W. K. "Low-complexity data reusing methods in adaptive filtering", *Signal Processing, IEEE Transactions on*, v. 52, n. 2, pp. 394–405, fev. 2004. ISSN: 1053-587X. doi:10.1109/TSP.2003.821338. 155

[206] WANG, Z., BOVIK, A., SHEIKH, H., et al. "Image quality assessment: from error visibility to structural similarity", *Image Processing, IEEE Transactions on*, v. 13, n. 4, pp. 600–612, abr. 2004. ISSN: 1057-7149. doi:10.1109/TIP.2003.819861. 158

[207] ZUO, C., CHEN, Q., GU, G., et al. "Scene-based nonuniformity correction algorithm based on interframe registration", *Journal of the Optical Society of America*, v. 28, n. 6, pp. 1164–1176, jun. 2011. doi:10.1364/JOSAA.28.001164. Disponível em: <http://josaa.osa.org/abstract.cfm?URI=josaa-28-6-1164>. 159

[208] BROX, T., BRUHN, A., PAPENBERG, N., et al. "High Accuracy Optical Flow Estimation Based on a Theory for Warping". In: *8th European Conference on Computer Vision*, v. 3024, pp. 25–36, Prague, Czech Republic, maio 2004. doi:10.1007/978-3-540-24673-2_3. Disponível em: <http://lmb.informatik.uni-freiburg.de/Publications/2004/Bro04a/brox_eccv04_of.pdf>. 161, 162

[209] ANANDAN, P. "A Computational Framework and an Algorithm for the Measurement of Visual Motion", *International Journal of Computer Vision*, v. 2, n. 3, pp. 283–310, 1989.

[210] SIMONCELLI, E. P., ADELSON, E. H., HEEGER, D. J. "Probability Distributions of Optical Flow". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 310–315, Mauii, Hawaii, jun. 1991.

[211] SIMONCELLI, E. P. "Coarse-to-fine Estimation of Visual Motion". In: *Eighth Workshop on Image and Multidimensional Signal Processing*, pp. 128–129, Cannes France, set. 1993. 162

[212] CAIN, S., HAYAT, M., ARMSTRONG, E. "Projection-based image registration in the presence of fixed-pattern noise", *Image Processing, IEEE Transactions on*, v. 10, n. 12, pp. 1860–1872, dez. 2001. ISSN: 1057-7149. doi:10.1109/83.974571. 162, 166

[213] TSAI, R., HUANG, T. "Multipleframe image restoration and registration", *Advances in Computer Vision and Image Processing*, v. 1, pp. 317–339, 1984. 169, 172

[214] KIM, S., BOSE, N., VALENZUELA, H. "Recursive reconstruction of high resolution image from noisy undersampled multiframes", *Acoustics, Speech and Signal Processing, IEEE Transactions on*, v. 38, n. 6, pp. 1013–1027, jun. 1990. ISSN: 0096-3518. doi:10.1109/29.56062.

[215] BOSE, N., KIM, H., VALENZUELA, H. "Recursive implementation of total least squares algorithm for image reconstruction from noisy, undersampled multiframes". In: *IEEE Conf. Acoustics, Speech and Signal Processing*, v. 5, pp. 269–272, Minneapolis, MN, abr. 1993.

[216] KIM, S., SU, W.-Y. "Recursive high-resolution reconstruction of blurred multiframe images", *Image Processing, IEEE Transactions on*, v. 2, n. 4, pp. 534–539, out. 1993. ISSN: 1057-7149. doi:10.1109/83.242363.

[217] RHEE, S., KANG, M. "Discrete cosine transform based regularized high-resolution image reconstruction algorithm", *Optical Engineering*, v. 38, n. 8, pp. 1348–1356, ago. 1999. 169

[218] ALAM, M., BOGNAR, J., HARDIE, R., et al. "Infrared image registration and high-resolution reconstruction using multiple translationally shifted aliased video frames", *Instrumentation and Measurement, IEEE Transactions on*, v. 49, n. 5, pp. 915–923, out. 2000. ISSN: 0018-9456. doi:10.1109/19.872908. 171

[219] KALTENBACHER, E., HARDIE, R. C. "High resolution infrared image reconstruction using multiple, low resolution, aliased frames". In: *Aerospace and Electronics Conference, 1996. NAECON 1996., Proceedings of the IEEE 1996 National*, v. 2, pp. 702–709, maio 1996. 171

[220] KOMATSU, T., AIZAWA, K., IGARASHI, T., et al. "Signal-processing based method for acquiring very high resolution images with multiple cameras and its theoretical analysis", *Communications, Speech and Vision, IEE Proceedings I*, v. 140, n. 1, pp. 19–24, fev. 1993. 172

[221] SAITO, T., KOMATSU, T., AIZAWA, K. "A signal-processing based method for acquiring very high resolution images with multiple cameras and its theoretical analysis". In: *Image Processing and its Applications, 1992., International Conference on*, pp. 365–368, abr. 1992. 172

[222] LANDWEBER, L. "An iteration formula for Fredholm integral equations of the first kind", *Amer. J. Math.*, v. 73, pp. 615–624, 1951. 172

[223] NGUYEN, N., MILANFAR, P. "An efficient wavelet-based algorithm for image superresolution". In: *Image Processing, 2000. Proceedings. 2000 International Conference on*, v. 2, pp. 351–354, set. 2000. 172, 174

[224] NGUYEN, N., MILANFAR, P. "A wavelet-based interpolation-restoration method for superresolution", *Circuits, Systems, and Signal Processing*, v. 19, n. 4, pp. 321–338, jul. 2000. 172, 174

[225] VAIDYANATHAN, P. P., DJOKOVIC, I. *The Circuits and Filters Handbook*. Boca Raton, Florida, CRC Press, 2003. Chapter 6: Wavelet Transforms. 172

[226] STARK, H., OSKOUI, P. "High-resolution image recovery from image-plane arrays using convex projections", *Journal of the Optical Society of America A*, v. 6, n. 11, pp. 1715–1726, nov. 1989. 174, 175

[227] TEKALP, A. M., OZKAN, M. K., SEZAN, M. I. "High-resolution image reconstruction from lower-resolution image sequences and space varying image restoration". In: *IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, v. 3, pp. 169–172, San Francisco, CA, mar. 1992. 175

[228] PATTI, A. J., SEZAN, M. I., TEKALP, A. M. "Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time", *Image Processing, IEEE Transactions on*, v. 6, n. 8, pp. 1064–1076, ago. 1997. ISSN: 1057-7149. doi:10.1109/83.605404. 175

[229] PATTI, A. J., ALTUNBASAK, Y. "Artifact reduction for set theoretic super resolution image reconstruction with edge adaptive constraints and higher-order interpolants", *Image Processing, IEEE Transactions on*, v. 10, n. 1, pp. 179–186, jan. 2001. ISSN: 1057-7149. doi:10.1109/83.892456. 175

[230] ALTUNBASAK, Y., PATTI, A. J., MERSEREAU, R. M. "Super-resolution still and video reconstruction from MPEG-coded video", *Circuits and Systems for Video Technology, IEEE Transactions on*, v. 12, n. 4, pp. 217–226, abr. 2002. ISSN: 1051-8215. doi:10.1109/76.999200. 175

[231] PELEG, S., KEREN, D., SCHWEITZER, L. "Improving image resolution using subpixel motion", *Pattern recognition letters*, v. 5, n. 3, pp. 223–226, 1987. ISSN: 0167-8655. 175

[232] IRANI, M., PELEG, S. "Super resolution from image sequences". In: *Pattern Recognition, 1990. Proceedings., 10th International Conference on*, v. ii, pp. 115–120 vol.2, jun. 1990. 176

[233] HARDIE, R. C., BARNARDB, K. J., BOGNARC, J. G., et al. "High Resolution Image Reconstruction From a Sequence of Rotated and Translated Frames and its Application to an Infrared Imaging System", *Optical Engineering*, v. 37, n. 1, pp. 247–260, jan. 1997. 176

[234] ELAD, M., FEUER, A. "Superresolution restoration of an image sequence: adaptive filtering approach", *Image Processing, IEEE Transactions on*, v. 8, n. 3, pp. 387–395, mar. 1999. ISSN: 1057-7149. doi:10.1109/83.748893. 176, 177

[235] ELAD, M., FEUER, A. "Super-resolution reconstruction of image sequences", *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 21, n. 9, pp. 817–834, set. 1999. ISSN: 0162-8828. doi:10.1109/34.790425. 176, 177

[236] NGUYEN, N., MILANFAR, P., GOLUB, G. "A computationally efficient superresolution image reconstruction algorithm", *Image Processing, IEEE Transactions on*, v. 10, n. 4, pp. 573–583, abr. 2001. ISSN: 1057-7149. doi:10.1109/83.913592. 177

[237] BORMAN, S., STEVENSON, R. *Spatial Resolution Enhancement of Low-Resolution Image Sequences - A Comprehensive Review with Directions for Future Research*. Relatório técnico, Laboratory for Image and Signal Analysis (LISA), University of Notre Dame, 1998. 177

[238] SCHULTZ, R. R., STEVENSON, R. L. "Extraction of high-resolution frames from video sequences", *Image Processing, IEEE Transactions on*, v. 5, n. 6, pp. 996–1011, jun. 1996. ISSN: 1057-7149. doi:10.1109/83.503915. 178

[239] HARDIE, R. C., BARNARD, K. J., , et al. "Joint MAP registration and high-resolution image estimation using a sequence of undersampled images", *Image Processing, IEEE Transactions on*, v. 6, n. 12, pp. 1621–1633, dez. 1997. ISSN: 1057-7149. doi:10.1109/83.650116. 178

[240] DANIELYAN, A., FOI, A., KATKOVNIK, V., et al. "Image and video super-resolution via spatially adaptive block-matching filtering". In: *Int. Workshop on Local and Non-Local Approximation (LNLA2008)*, Lausanne, Switzerland, 2008. 180

[241] ELAD, M. "On the origin of the bilateral filter and ways to improve it", *Image Processing, IEEE Transactions on*, v. 11, n. 10, pp. 1141–1151, out. 2002. ISSN: 1057-7149. doi:10.1109/TIP.2002.801126. 181

[242] GILBOA, G., OSHER, S. "Nonlocal linear image regularization and supervised segmentation", *Multiscale Modeling & Simulation*, v. 6, n. 2, pp. 595–630, 2007. 181

[243] TAKEDA, H., FARSIU, S., MILANFAR, P. "Kernel Regression for Image Processing and Reconstruction", *Image Processing, IEEE Transactions on*, v. 16, n. 2, pp. 349–366, fev. 2007. ISSN: 1057-7149. doi:10.1109/TIP.2006.888330. 184

[244] PUN, W., JEFFS, B. "Shape parameter estimation for generalized Gaussian Markov random field models used in MAP image restoration". In: *Signals, Systems and Computers, 1995. 1995 Conference Record of the Twenty-Ninth Asilomar Conference on*, v. 2, pp. 1472 –1476 vol.2, nov. 1995. doi:10.1109/ACSSC.1995.540942. 185

[245] AIAZZI, B., ALPARONE, L., BARONTI, S. "Estimation based on entropy matching for generalized Gaussian PDF modeling", *Signal Processing Letters, IEEE*, v. 6, n. 6, pp. 138–140, jun. 1999. ISSN: 1070-9908. doi:10.1109/97.763145.

[246] DOMINGUEZ-MOLINA, J., GONZÁLEZ-FARÍAS, G., RODRÍGUEZ-DAGNINO, R., et al. *A practical procedure to estimate the shape parameter in the generalized Gaussian distribution*. Relatório técnico, CIMAT, 2001. Disponível em: <http://www.cimat.mx/reportes/enlinea/I-01-18_eng.pdf>. 185

[247] TOMASI, C., MANDUCHI, R. "Bilateral filtering for gray and color images". In: *IEEE Int. Conf. Computer Vision*, pp. 836–846, New Delhi, India, 1998. 192