1 Acronyms

ADMM: Alternating Directions Method of Multipliers

ISTA: Iterative Shrinkage Thresholding Algorithm

FISTA: Fast Iterative Shrinkage Tresholding Algorithm

RGB image: an image with red, blue, and green channels

JPEG: the compression process developed by the Joint Photographic Experts Group

2 Matrices

Matrices are noted as bold capital letters: $\boldsymbol{D}$, $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{\Phi}$, $\boldsymbol{S}$, $\boldsymbol{\ss}$, $\boldsymbol{X}$, $\boldsymbol{U}$, $\boldsymbol{V}$.

$\boldsymbol{D}$ is the a dictionary. For most of the dissertation, $\boldsymbol{D}$ has circulant matrix blocks.

$\boldsymbol{S}$ is a collection of signal vectors, either gathering multiple channels or multiple samples.

$\boldsymbol{X}$ is a collection of dictionary coefficient vectors, corresponding to multple signal vectors.

$\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{U}$, and $\boldsymbol{V}$ are arbitrary matrices. $\boldsymbol{A}$ and $\boldsymbol{B}$ are also used as the linear operators in the ADMM contraints.

$\boldsymbol{\Phi}$ is an arbitrary linear operator (this matrix is also listed under operators, since matrices are linear operators).

$\boldsymbol{T}$ a diagonal matrix that has diagonal elements of 1 for dictionary elements that are not constrained to be zero, and zeros for the other diagonal elements (this matrix is also listed under operators since matrices are linear operators).

$\boldsymbol{W}$ converts from RGB to YUV, downsamples the UV channels, and computes the DCT of 8×8 blocks. Naturally, this matrix also appears under operators.

$\boldsymbol{\ss}$ is part of the dictionary for the product dictionary model.

$\boldsymbol{Q}$ is used to represent the matrix $\rho\mathbf{I} + \hat{\boldsymbol{D}}^H\hat{\boldsymbol{D}}$

$\boldsymbol{\Xi}$ is used to represent the matrix $\rho\mathbf{I} + \hat{\boldsymbol{D}}\hat{\boldsymbol{D}}^H$

3 vectors

Vectors are bold and lower-case.

$\boldsymbol{x}$, $\boldsymbol{y}$ are the primal variables for ADMM.

$\boldsymbol{u}$ is the dual variable for ADMM.

$\boldsymbol{c}$ is the constraint vector in ADMM.

$\boldsymbol{x}$, $\boldsymbol{z}$ are the coefficients for dictionary model in ADMM algorithm.

$\boldsymbol{v}$ is another primal variable (grouped with $\boldsymbol{z}$) used in chapter 4.

$\boldsymbol{\gamma}$ is the corresponding dual variable.

$\boldsymbol{x}$ and $\boldsymbol{z}$ are also used as vectors in the FISTA algorithm.

$\boldsymbol{x}$ is also used as an arbitrary vector throughout document. Context will make clear.

$\boldsymbol{s}$ is the signal.

$\boldsymbol{b}$ is another arbitrary vector.

$\boldsymbol{u}$ and $\boldsymbol{v}$ are more arbitrary vectors, usually used in pairs. The may collectively specify a rank-1 update to a matrix: $\boldsymbol{u}\boldsymbol{v}^H$.

$\boldsymbol{f}$ is a dictionary filter.

$\boldsymbol{d}$ is a column of $\boldsymbol{D}$.

$\boldsymbol{q}$ is the quality-factor dependent vector used in quantization in JPEG compression.

$\boldsymbol{\omega}$ is an eigenvector.

$\boldsymbol{R}$ is a rescaling matrix, used to scale a normalized dictionary back to its unnormalized form.

4 Non-integer Scalars

Non-integer scalars are usually lower-case script letters that are not bolded. (The exception is the estimate of the Lipshitz constant used to determine stepsize in ISTA and FISTA.) I say "non-integer" not imply that they cannot take on integer values, but merely to differentiate them from the scalars that are required to be nonnegative integers.

a and b are arbitrary scalars.

ρ is a scalar for the ADMM alrogithm that specifies both the weighting of the constraints in the augmented Lagrangian and the stepsize in the dual variable update.

α is the over-relaxation or under-relaxation factor for ADMM

λ is the factor for L_1 penalty. λ is also used as the factor for the L_2 penalty on the image gradients for Tikhonov regularization. Context will make clear.

$\mathbb{L}$ is an estimate of the Lipshitz constant, used to determine stepsize in the ISTA and FISTA algorithms.

τ is the eigenvalue

r and ω are used to specify momentum stepsize in FISTA and a FISTA-like algorithm. Always appear with superscripts specifying iteration.

$\mathscr{L}$ the loss (as in the loss function)

5 Indexing integers

n selects the sample

t selects the iteration

m selects the filter

c selects the channel

$\hat{k}$ specifies the frequency

6 Integer Constants

M is the number of filters

C is the number of channels

$\hat{K}$ is the number of elements in a single channel of the signal

Я is a small integer that specifies the rank of the dictionary updates

L is the number of layers

7 functions and operations

$*$ is used to mean circular convolution, except when discussing boundary handling, which works to make circular convolution and convolution equivalent.

$\cdot^*$ is the complex conjugate.

$\cdot^H$ is the Hermitian transpose of a vector or matrix

$\mathcal{L}_\rho$ is the augmented Lagrangian function

S is the shrinkage operator

$\mathcal{F}$ applies the Fourier tranform to each channel and/or filter

f and g are arbitrary convex functions. f may also be used to specify the objective function of a minimization problem. f and g are also used for arbitrary functions that are part of a composite function. Context should make clear.

$\mathrm{q}(\cdot)$ quantizes a vector.

$\mathbb{1}_{\text{condition}}$ takes on a value of 0 when the condition is true and ∞ when the condition is false.

$\arg\min$ is the argument minimum of a function

$\nabla_a b$ is the gradient of b in respect to a.

L_1 is the L_1 norm

L_2 is the L_2 norm

$\mathbf{\Phi}$ is an arbitrary linear operator (this operator is also listed under matrices, since matrices are linear operators).

$\boldsymbol{T}$ zeros out all dictionary elements that are constrained to be zero. (This operator is also listed under matrices since matrices are linear operators.)

$\boldsymbol{W}$ converts from RGB to YUV, downsamples the UV channels, and computes the DCT of 8×8 blocks. (This operator also appears under matrices.)

8 Superscripts

Conventionally, superscripts are used to indicate exponents.

However, I also use superscripts for other purposes. To distinguish these superscripts from exponents, I put them in parenthesis $\boldsymbol{x}^{(\cdot)}$.

Superscripts are used to indicate which signal sample (or the corresponding dictionary update).

Superscripts are also used to specify the iteration number.

Finally, superscripts are used on gradients to specify a particular gradient term.

9 Subscripts

subscript m specifies the filter. If there are multiple layers, $[m]$ will be used instead of subscript m.

subscript n specifies the sample.

Subscript c specifies the channel. If there are multiple layers $[c]$ will be used instead of subscript c.

subscript ℓ specifies the layer.

subscripts $+$ and $-$ are used to specify the eigenvalues or eigenvectors of a 2×2 matrix corresponding to the plus or minus in the quadratic formula.

The ρ in $\mathcal{L}_\rho$ specifies the scalar weight of the L_2 norm related to the affine constraints, used in the augmented Lagrangian function.

Subscript $\cdot_{\text{init}}$ is short for initial value.

Subscript $\cdot_{\text{sc}}$ is short for "scaled", and indicates that the variable in the algorithm is a scaled form of the variable.

Subscript i is used for essentially all other indexing.