

Dual-Tree Complex Wavelets - their key properties and a range of image-processing applications

Nick Kingsbury

Signal Processing and Communications Laboratory
Department of Engineering, University of Cambridge, UK.

email: ngk@eng.cam.ac.uk web: www.eng.cam.ac.uk/~ngk

Part 1, UDRC Short Course, 21 March 2013



UNIVERSITY OF
CAMBRIDGE

Complex Wavelets: What are they and what can they do?

- Basic form of the DT CWT
- Shift invariance of subband transfer functions
- DT CWT in 2-D – directional selectivity (6 subbands)
- DT CWT in 3-D – more selectivity (28 subbands)
- Image Registration
- Fusion
- Deconvolution with sparsity-based regularisation

Features of the (Real) Discrete Wavelet Transform (DWT)

- **Good compression** of signal energy.
- **Perfect reconstruction** with short support filters.
- **No redundancy** – hence orthonormal or bi-orthogonal transforms are possible.
- **Very low computation** – order- N only.

But

- **Severe shift dependence.**
- **Poor directional selectivity** in 2-D, 3-D etc.

The DWT is normally implemented with a tree of highpass and lowpass filters, separated by 2 : 1 decimators.

Real Discrete Wavelet Transform (DWT) in 1-D

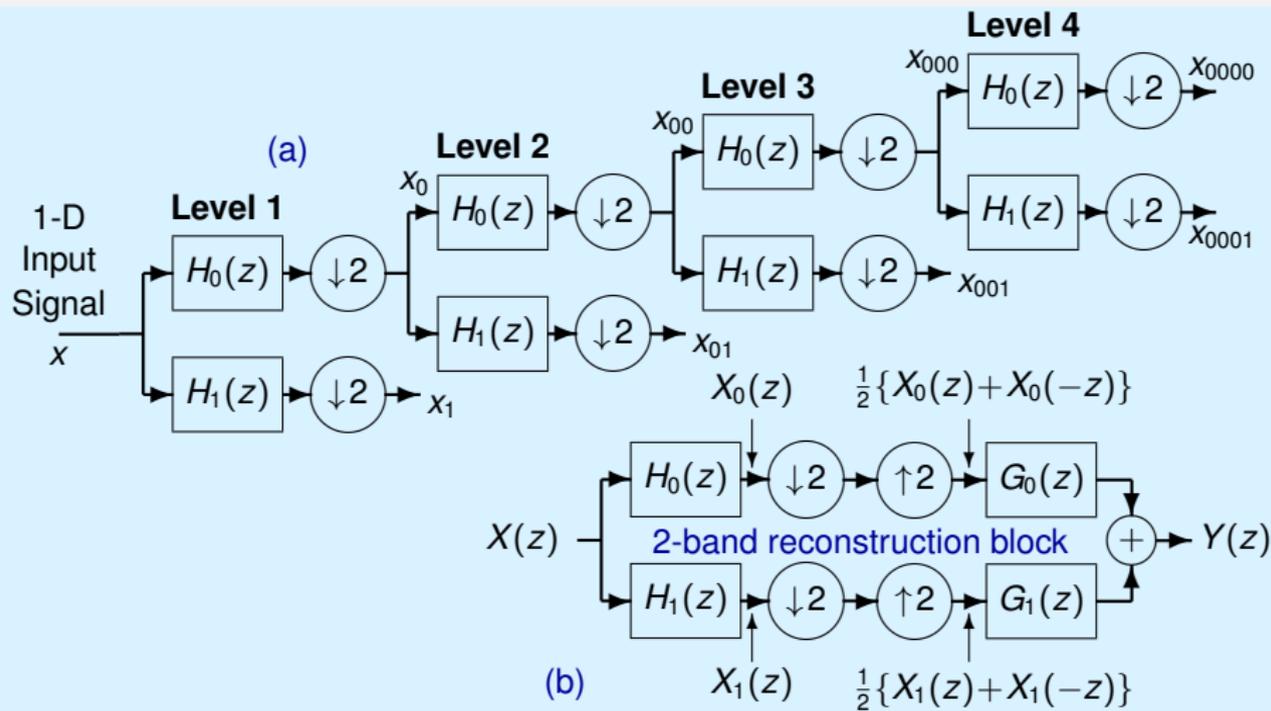


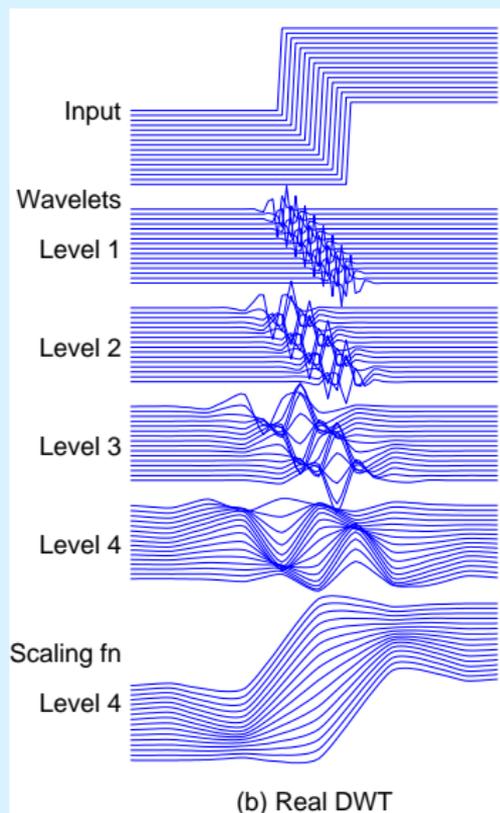
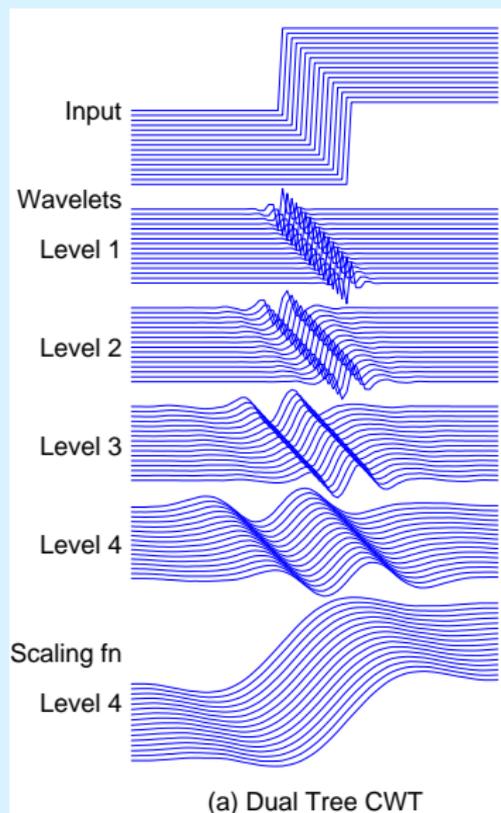
Figure: (a) Tree of real filters for the DWT. (b) Reconstruction filter block for 2 bands at a time, used in the inverse transform.

Visualising Shift Invariance / Dependence

- Apply a standard input (e.g. unit step) to the transform for a **range of shift positions**.
- Select the transform coefficients from **just one wavelet level** at a time.
- Inverse transform each set of selected coefficients.
- Plot the component of the reconstructed output for each shift position at each wavelet level.
- Check for **shift invariance** (similarity of waveforms).

See Matlab demonstration / next slide.

Shift Invariance of DT CWT / Dependence of DWT



Features of the Dual Tree Complex Wavelet Transform (DT CWT)

- Good **shift invariance** = **negligible aliasing**. Hence transfer function through each subband is independent of shift **and** wavelet coefs can be interpolated within each subband, independent of all other subbands.
- Good **directional selectivity** in 2-D, 3-D etc. – derives from **analyticity** in 1-D (ability to separate positive from negative frequencies).
- **Perfect reconstruction** with short support filters.
- **Limited redundancy** – 2:1 in 1-D, 4:1 in 2-D etc.
- **Low computation** – much less than the undecimated (à trous) DWT.

Each tree contains purely real filters, but the two trees produce the **real and imaginary parts** respectively of each complex wavelet coefficient.

Q-shift Dual Tree Complex Wavelet Transform in 1-D

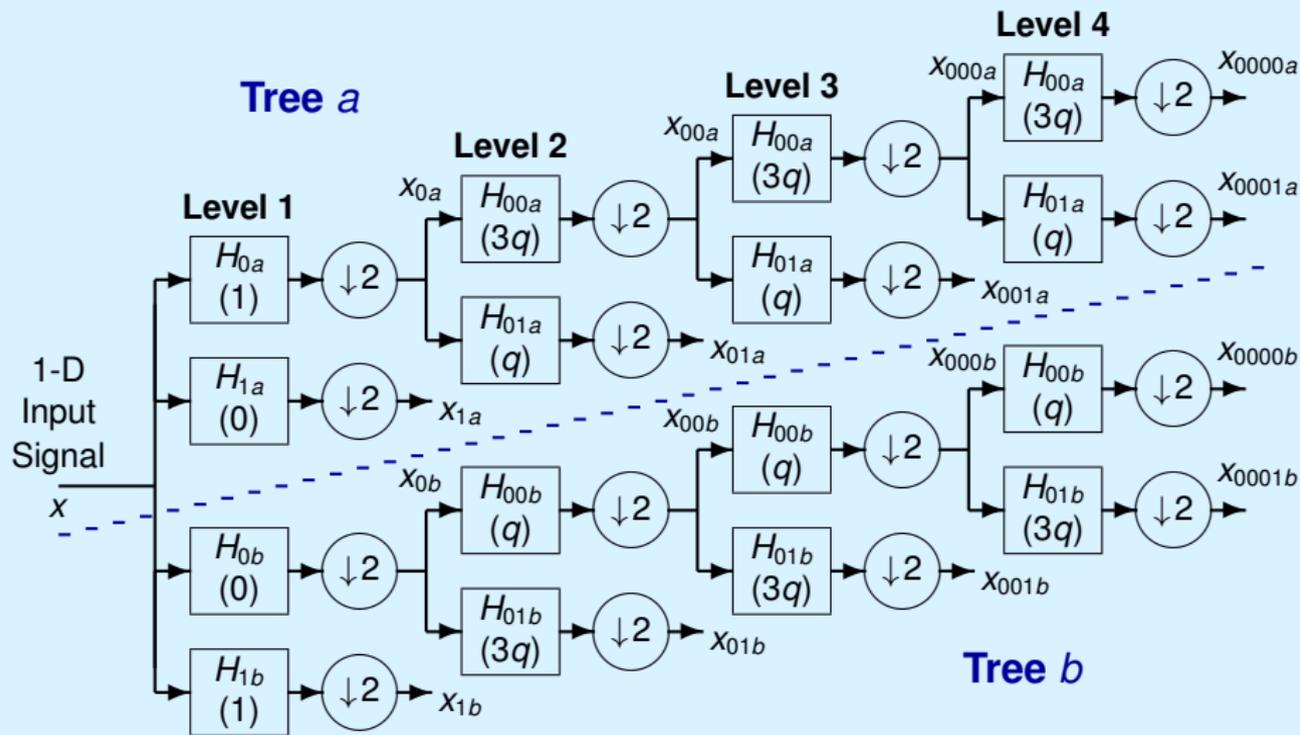


Figure: Dual tree of real filters for the Q-shift CWT, giving real and imaginary parts of complex coefficients from tree a and tree b respectively.

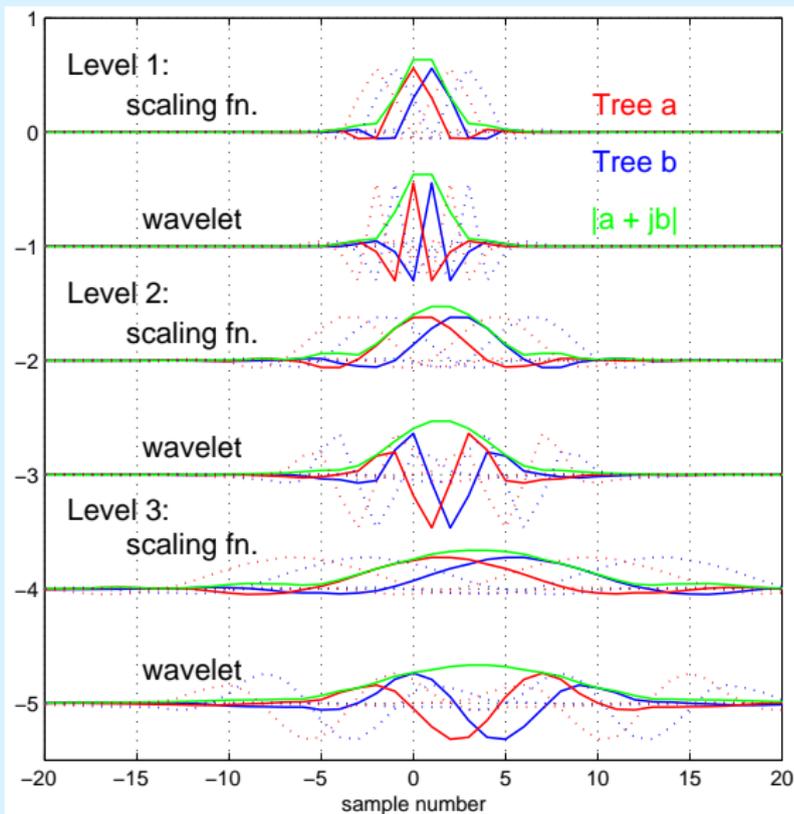
Features of the Q-shift Filters

Below level 1:

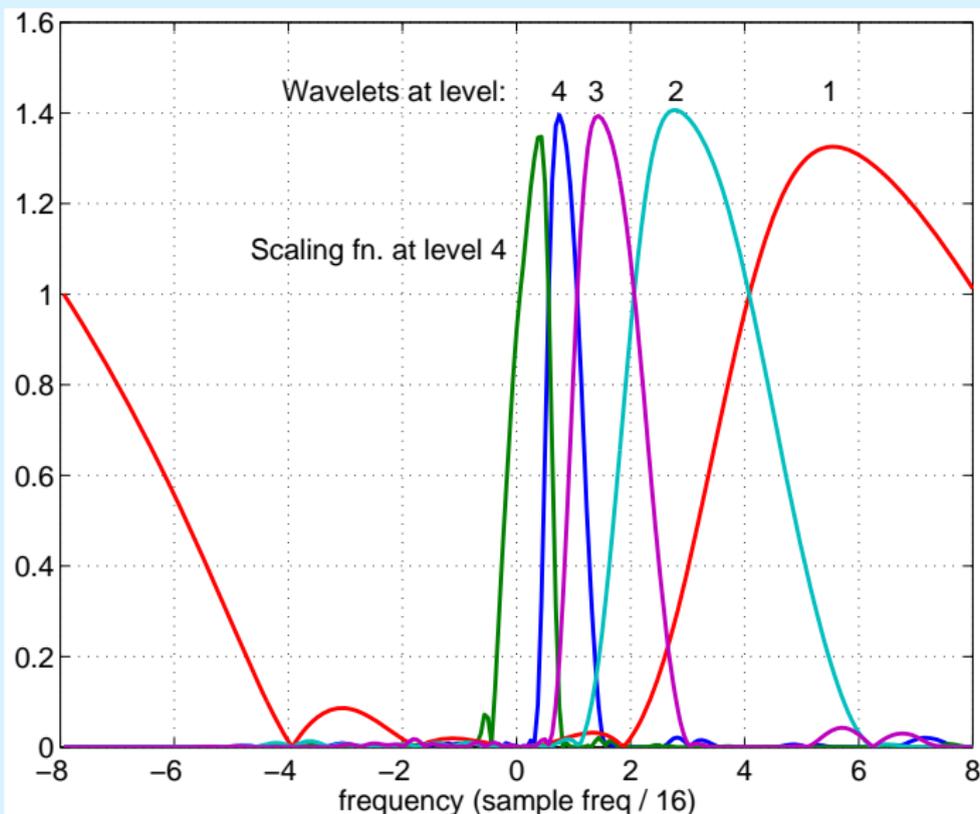
- Half-sample delay difference is obtained with filter delays of $\frac{1}{4}$ and $\frac{3}{4}$ of a sample period (instead of 0 and $\frac{1}{2}$ a sample for our original DT CWT).
- This is achieved with an **asymmetric even-length** filter $H(z)$ and its time reverse $H(z^{-1})$.
- Due to the asymmetry (like Daubechies filters), these may be designed to give an **orthonormal perfect reconstruction** wavelet transform.
- Tree **b** filters are the **reverse** of tree **a** filters, and reconstruction filters are the reverse of analysis filters, so **all filters** are from the **same orthonormal set**.
- Both trees have the **same frequency responses**.
- The combined **complex** impulse responses are **conjugate symmetric** about their mid points, even though the separate responses are asymmetric. Hence **symmetric extension** still works at image edges.

Q-shift DT CWT Basis Functions – Levels 1 to 3

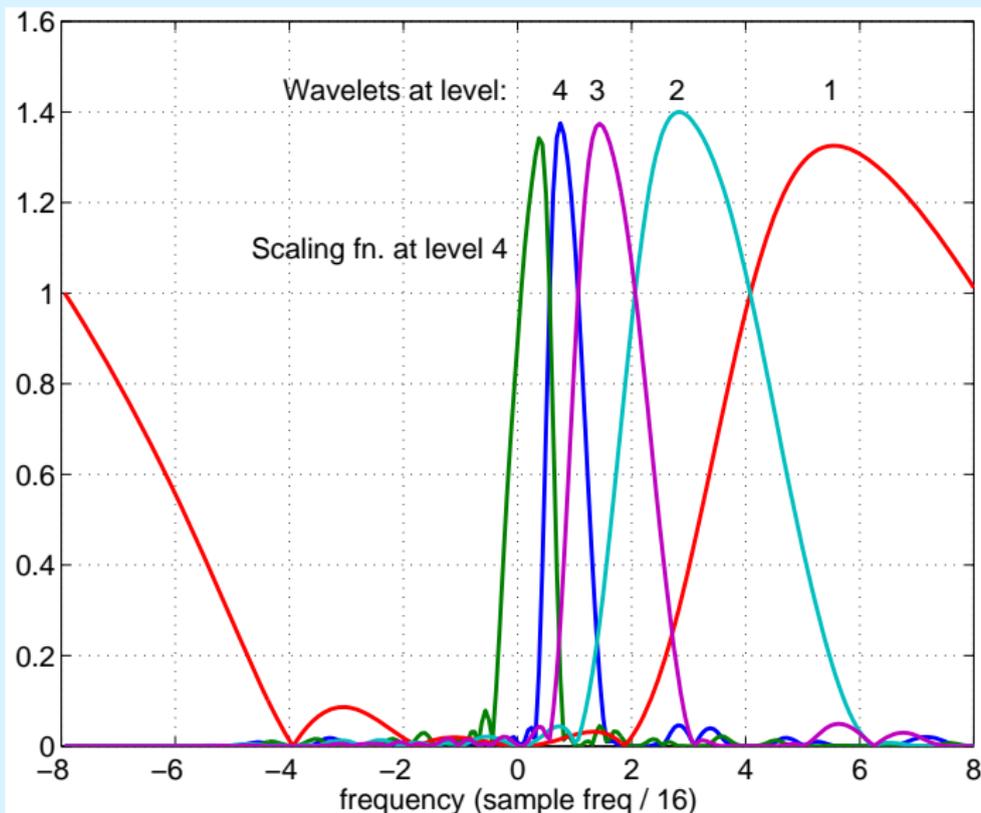
Basis functions for adjacent sampling points are shown dotted.



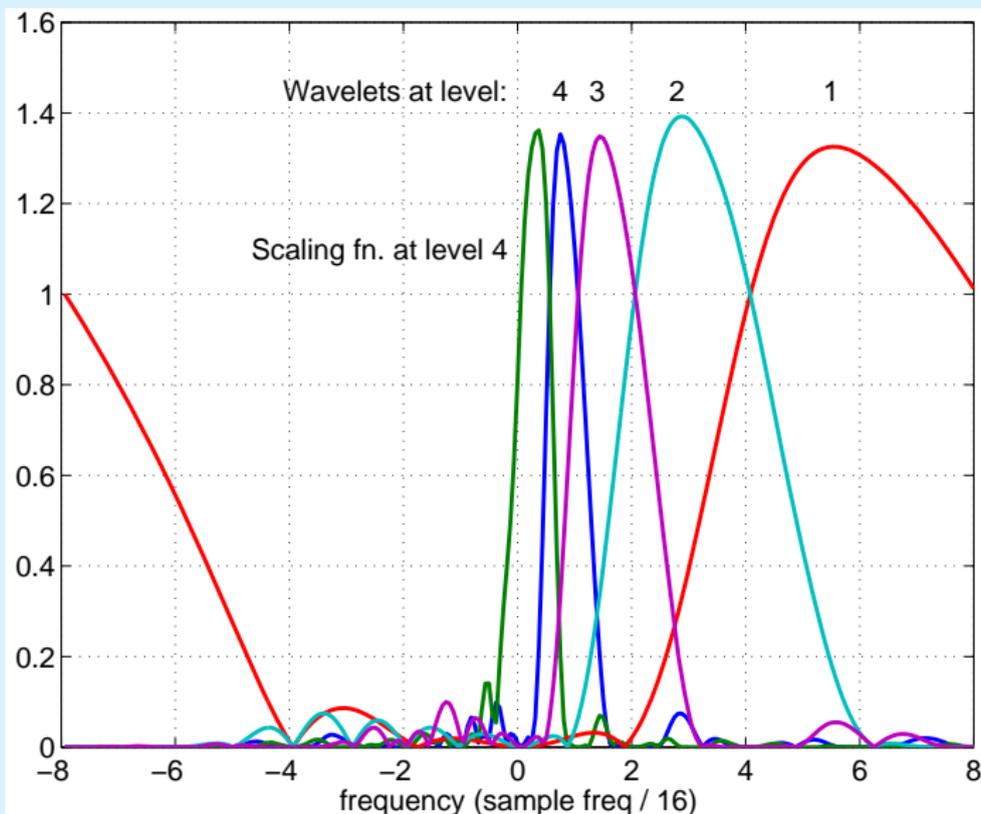
Frequency Responses of 18-tap Q-shift filters



Frequency Responses of 14-tap Q-shift filters



Frequency Responses of 6-tap Q-shift filters

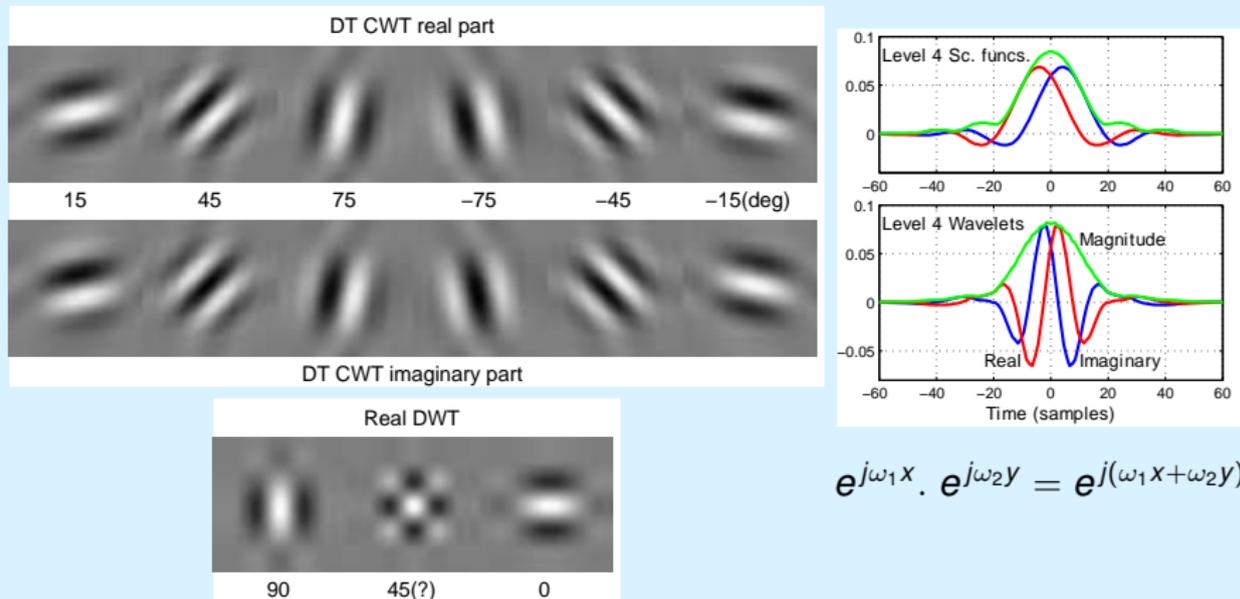


The DT CWT in 2-D

When the DT CWT is applied to 2-D signals (images), it has the following features:

- It is performed separably, using 2 trees for the rows of the image and 2 trees for the columns – yielding a **Quad-Tree** structure (4:1 redundancy).
- The 4 quad-tree components of each coefficient are combined by simple sum and difference operations to yield a **pair of complex coefficients**. These are part of two separate subbands in adjacent quadrants of the 2-D spectrum.
- This produces **6 directionally selective subbands** at each level of the 2-D DT CWT. Fig 3 shows the basis functions of these subbands at level 4, and compares them with the 3 subbands of a 2-D DWT.
- The DT CWT is directionally selective (see fig 5) because the complex filters can **separate positive and negative frequency components** in 1-D, and hence **separate adjacent quadrants** of the 2-D spectrum. Real separable filters cannot do this!

2-D Basis Functions at level 4

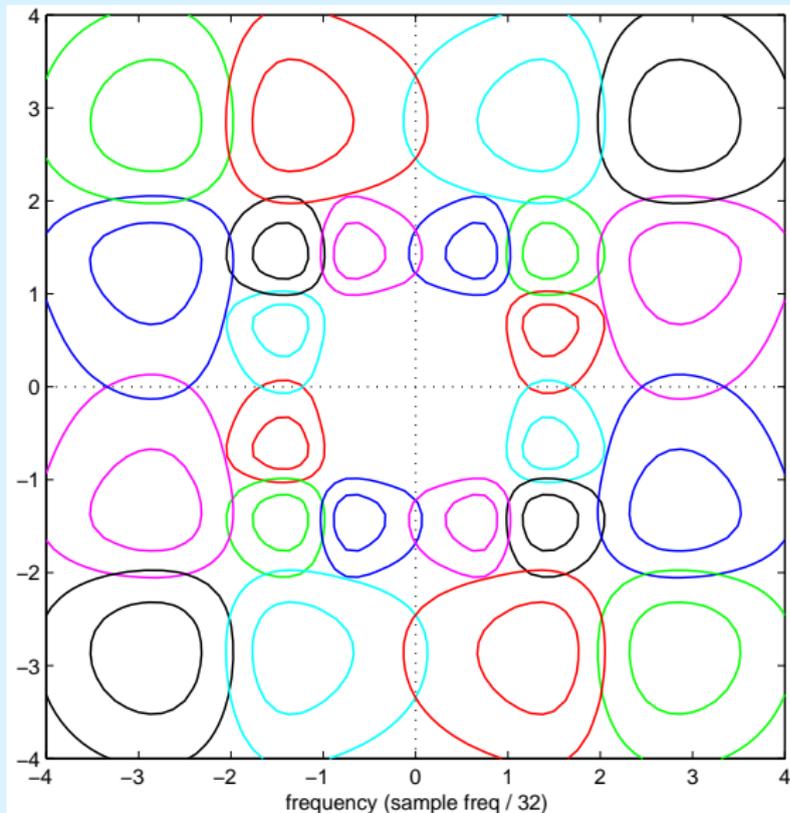


$$e^{j\omega_1 x} \cdot e^{j\omega_2 y} = e^{j(\omega_1 x + \omega_2 y)}$$

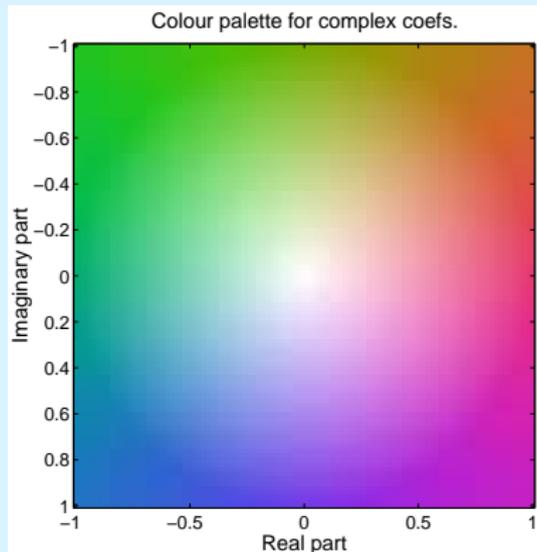
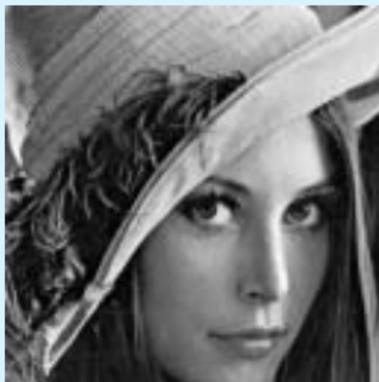
Figure: Basis functions of 2-D Q-shift complex wavelets (top), and of 2-D real wavelet filters (bottom), all illustrated at level 4 of the transforms. The complex wavelets provide 6 directionally selective filters, while real wavelets provide 3 filters, only two of which have a dominant direction. The 1-D bases, from which the 2-D complex bases are derived, are shown to the right.

Frequency Responses of 2-D Q-shift filters at levels 3 and 4

Contours shown at
-1 dB and -3 dB.



Test Image and Colour Palette for Complex Coefficients



2-D DT CWT Decomposition into Subbands

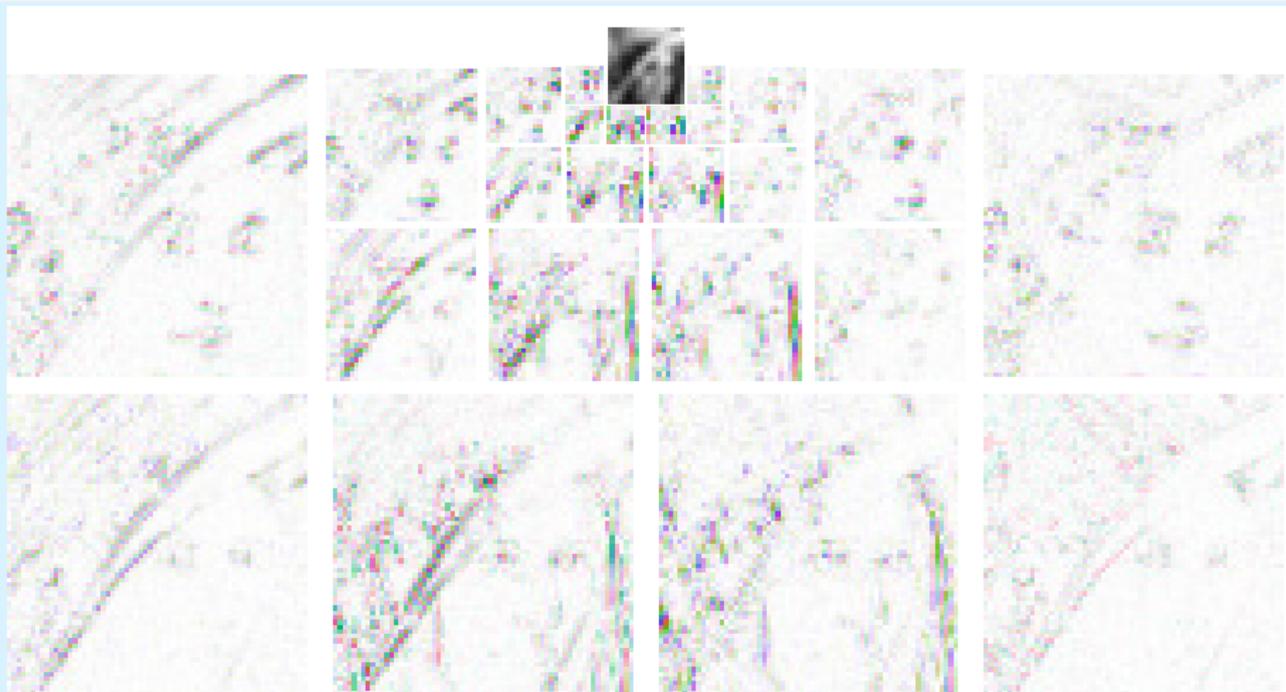


Figure: Four-level DT CWT decomposition of **Lena** into 6 subbands per level (only the central 128×128 portion of the image is shown for clarity). A colour-wheel palette is used to display the complex wavelet coefficients.

2-D DT CWT reconstruction components from each subband

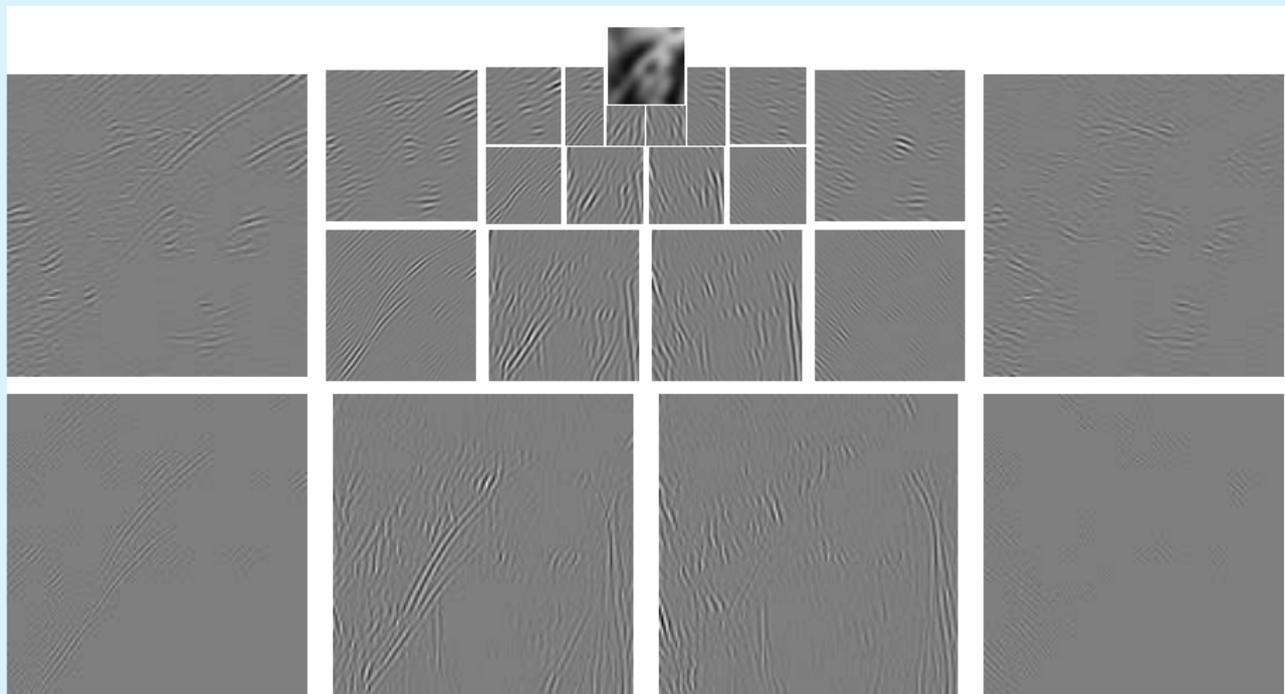


Figure: Components from each subband of the reconstructed output image for a 4-level DT CWT decomposition of **Lena** (central 128×128 portion only).

2-D Shift Invariance of DT CWT vs DWT

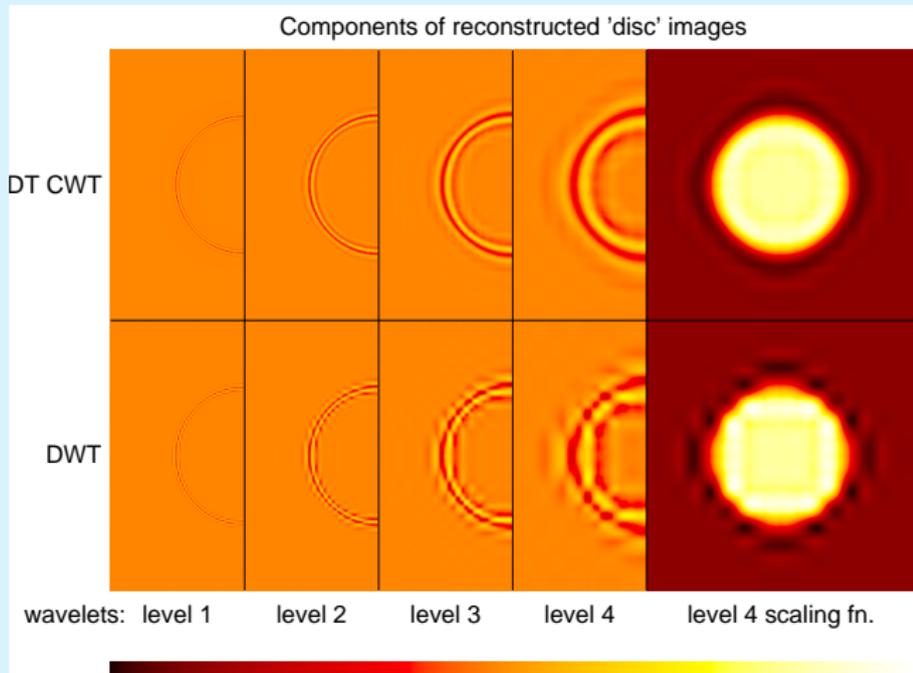
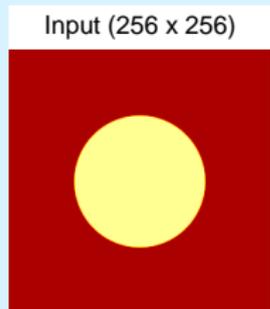


Figure: Wavelet and scaling function components at levels 1 to 4 of an image of a light circular disc on a dark background, using the 2-D DT CWT (upper row) and 2-D DWT (lower row). Only half of each wavelet image is shown in order to save space.

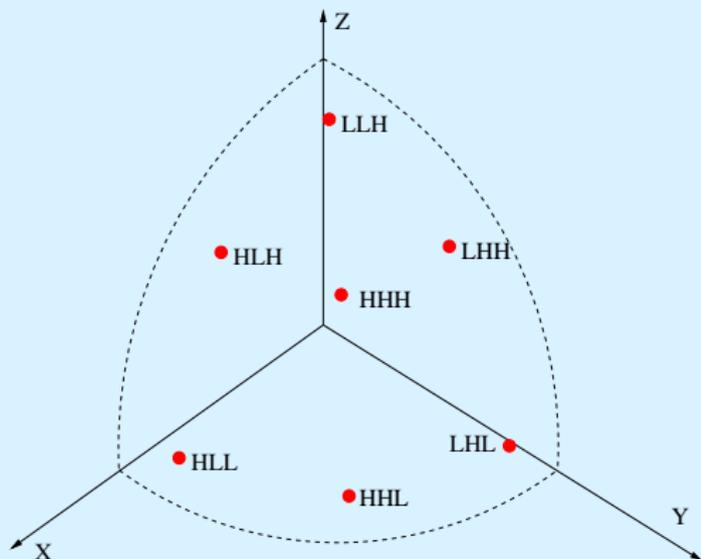
The DT CWT in 3-D

When the DT CWT is applied to 3-D signals (eg medical MRI or CT datasets), it has the following features:

- It is performed separably, with 2 trees used for the rows, 2 trees for the columns and 2 trees for the slices of the 3-D dataset – yielding an **Octal-Tree** structure (8:1 redundancy).
- The 8 octal-tree components of each coefficient are combined by simple sum and difference operations to yield a **quad of complex coefficients**. These are part of 4 separate subbands in adjacent octants of the 3-D spectrum.
- This produces **28 directionally selective subbands** ($4 \times (8 - 1)$) at each level of the 3-D DT CWT. The subband basis functions are now **planar waves** of the form $e^{j(\omega_1 x + \omega_2 y + \omega_3 z)}$, modulated by a 3-D Gaussian envelope.
- Each subband responds to approximately flat surfaces of a particular orientation. There are 7 orientations on each quadrant of a hemisphere.

3D subband orientations on one quadrant of a hemisphere

One octant of the
3D frequency domain:



3D Gabor-like basis functions:

$$h_{k_1, k_2, k_3}(x, y, z) \simeq e^{-(x^2 + y^2 + z^2)/2\sigma^2} \times e^{j(\omega_{k_1} x + \omega_{k_2} y + \omega_{k_3} z)}$$

These are **28 planar waves** (7 per quadrant of a hemisphere)
whose orientation depends on $\omega_{k_1} \in \{\omega_L, \omega_H\}$ and $\omega_{k_2}, \omega_{k_3} \in \{\pm\omega_L, \pm\omega_H\}$,
where $\omega_H \simeq 3\omega_L$.

Applications of the DT CWT

- **Motion estimation** [Magarey 98] and **compensation**
- **Registration** [Kingsbury 02, Chen 12]
- **Denoising** [Choi 00, Miller 06] and **deconvolution** [Jalobeanu 00, De Rivaz 01, J Ng 07, Y Zhang 10]
- **Texture analysis** [Hatipoglu 99] and **synthesis** [De Rivaz 00]
- **Segmentation** [De Rivaz 00, Shaffrey 02]
- **Classification** [Romberg 00] and **image retrieval** [Kam & T T Ng 00, Shaffrey 03]
- **Watermarking of images** [Loo 00] and **video** [Earl 03]
- **Compression / Coding** [Reeves 03]
- **Seismic analysis** [van Spaendonck & Fernandes 02, Miller 05]
- **Diffusion Tensor MRI visualisation** [Zymnis 04]
- **Object matching & recognition** [Anderson & Fauqueur 06, Nelson 11]

Application 1: Registration

Efficient displacement estimation / registration of noisy data

Applied to:

- Registration of medical datasets taken some time apart and correction for patient movement
- Conversion from low-quality video to high-quality still images – e.g. correction of fluctuations in atmospheric refraction (heat shimmer)
- Motion estimation for non-rigid objects and fluids
- Registration of multi-look images affected by speckle, usually due to illumination from coherent sources such as lasers or synthetic aperture radar (SAR).

Displacement estimation usually involves measuring **gradients, derivatives or differences**. High noise levels mean that registration algorithms must be **robust to noise** if the noise is uncorrelated between images.

Key Features of Robust Registration Algorithms

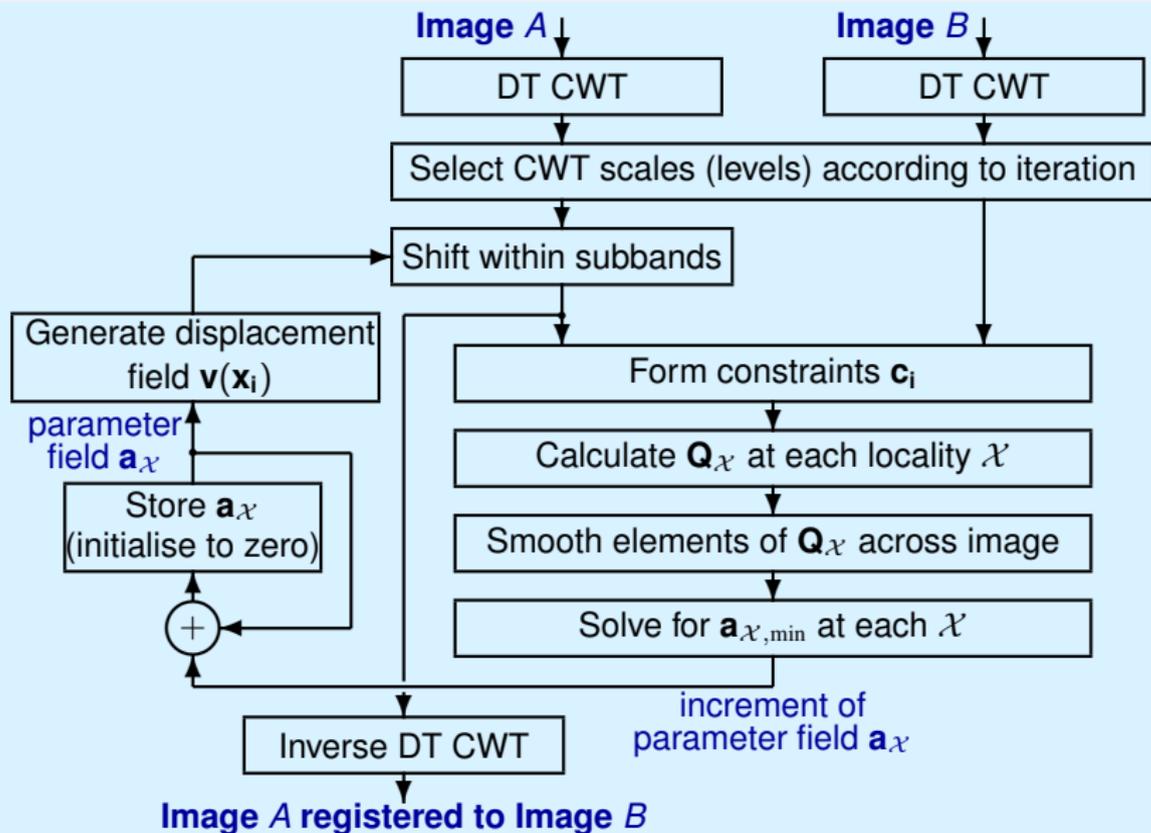
- Edge-based methods are more robust than point-based ones.
- Bandlimited multiscale (wavelet) methods allow spatially adaptive denoising.
- Phase-based bandpass methods can give rapid convergence and immunity to illumination changes between images (**but** we have to be careful about 2π ambiguities) .
- If the displacement field is smooth, a wider-area parametric (affine) model of the field is likely to be more robust than a highly-local translation-only model.

Note: Biological vision systems have evolved to use multiscale directional bandpass filters as their front-end process (e.g. the V1 cortical filters in humans / mammals).

Selected Methods

- Dual-tree Complex Wavelet Transform (DT CWT):
 - efficiently synthesises a multiscale directional shift-invariant filterbank, with perfect reconstruction;
 - provides complex coefficients whose phase shift depends approximately linearly on displacement;
 - allows each subband of coefficients to be interpolated (shifted) independently of other subbands (because of shift invariance of the subband $H(z)$).
- Parametric model of displacement field, whose solution is based on local edge-based motion constraints (Hemmendorff, Andersson, Kronander and Knutsson, IEEE Trans Medical Imaging, Dec 2002):
 - derives straight-line constraints from directional subbands of the DT CWT;
 - solves for spatially-varying affine model parameters which minimise constraint error energy over multiple directions and scales.

Registration Algorithm:



Basic Linear Flow Model

Key Assumption for local translation model:

- Time derivative of the phase θ of each complex wavelet coefficient depends **approximately linearly** on the local velocity vector \mathbf{v} .

This can be expressed as a flow equation in time and spatial derivatives:

$$\frac{\partial \theta}{\partial t} = \nabla_{\mathbf{x}} \theta \cdot \mathbf{v}$$

We can rearrange this to be in the form:

$$\nabla_{\mathbf{x}} \theta \cdot \mathbf{v} - \frac{\partial \theta}{\partial t} = 0$$

or

$$\begin{bmatrix} \nabla_{\mathbf{x}} \theta \\ -\frac{\partial \theta}{\partial t} \end{bmatrix}^T \tilde{\mathbf{v}} = 0 \quad \text{where} \quad \tilde{\mathbf{v}} = \begin{bmatrix} \mathbf{v} \\ 1 \end{bmatrix}$$

Parametric Model: Linear Constraint Equations

Let the displacement vector at the i^{th} location \mathbf{x}_i be $\mathbf{v}(\mathbf{x}_i)$; let $\tilde{\mathbf{v}}_i = \begin{bmatrix} \mathbf{v}(\mathbf{x}_i) \\ 1 \end{bmatrix}$.

Note that, as well as \mathbf{x}_i , the locator i also specifies a subband direction d_i ($1 \dots 6$) and a scale (level) s_i . A **straight-line constraint** on $\mathbf{v}(\mathbf{x}_i)$ can be written

$$\mathbf{c}_i^T \tilde{\mathbf{v}}_i = 0 \quad \text{or} \quad c_{1,i} v_{1,i} + c_{2,i} v_{2,i} + c_{3,i} = 0$$

For a phase-based system in which wavelet coefficients at $\{\mathbf{x}_i, d_i, s_i\}$ in images A and B have phases $\theta_{A,i}$ and $\theta_{B,i}$, approximate **linearity of phase θ vs. displacement $\mathbf{v}(\mathbf{x}_i)$** means that

$$\mathbf{c}_i^T \tilde{\mathbf{v}}_i \approx 0 \quad \text{if} \quad \mathbf{c}_i = C_i \begin{bmatrix} \nabla_{\mathbf{x}} \theta_i \\ \theta_{A,i} - \theta_{B,i} \end{bmatrix}$$

In practise we compute this by averaging finite differences at the centre \mathbf{x}_i of a $2 \times 2 \times 2$ block of coefficients from a given subband $\{d_i, s_i\}$ of images A and B .

Note: C_i is a constant which does not affect the line defined by the constraint, but it is important as a weight for combining constraint errors (see later).

Parameters of the Model

We can define a **6-term affine parametric model** \mathbf{a} for \mathbf{v} such that

$$\mathbf{v}(\mathbf{x}) = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} + \begin{bmatrix} a_3 & a_5 \\ a_4 & a_6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

or in a more useful form

$$\mathbf{v}(\mathbf{x}) = \begin{bmatrix} 1 & 0 & x_1 & 0 & x_2 & 0 \\ 0 & 1 & 0 & x_1 & 0 & x_2 \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ \vdots \\ a_6 \end{bmatrix} = \mathbf{K}(\mathbf{x}) \cdot \mathbf{a}$$

Affine models can synthesise translation, rotation, constant zoom, and shear. A **quadratic model**, which allows for linearly changing zoom (approx perspective), requires up to 6 additional parameters and columns in \mathbf{K} of the form

$$\begin{bmatrix} \dots & x_1 x_2 & 0 & x_1^2 & 0 & x_2^2 & 0 \\ \dots & 0 & x_1 x_2 & 0 & x_1^2 & 0 & x_2^2 \end{bmatrix}$$

Solving for the Model Parameters

Using techniques (due to Hemmendorff et al) similar to homogeneous coordinates:

$$\text{Let } \tilde{\mathbf{K}}_i = \begin{bmatrix} \mathbf{K}(\mathbf{x}_i) & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \text{ and } \tilde{\mathbf{a}} = \begin{bmatrix} \mathbf{a} \\ 1 \end{bmatrix} \text{ so that } \tilde{\mathbf{v}}_i = \tilde{\mathbf{K}}_i \tilde{\mathbf{a}}.$$

Ideally for a given scale-space locality \mathcal{X} , we wish to find the parametric vector $\tilde{\mathbf{a}}$ such that

$$\mathbf{c}_i^T \tilde{\mathbf{v}}_i = 0 \quad \text{when} \quad \tilde{\mathbf{v}}_i = \tilde{\mathbf{K}}_i \tilde{\mathbf{a}} \quad \text{for all } i \text{ such that } \{\mathbf{x}_i, d_i, s_i\} \in \mathcal{X}.$$

In practise this is an **overdetermined** set of equations, so we find the **LMS solution**, i.e. the value of \mathbf{a} which minimises the squared error

$$\mathcal{E}_{\mathcal{X}} = \sum_{i \in \mathcal{X}} \|\mathbf{c}_i^T \tilde{\mathbf{v}}_i\|^2 = \sum_{i \in \mathcal{X}} \|\mathbf{c}_i^T \tilde{\mathbf{K}}_i \tilde{\mathbf{a}}\|^2 = \sum_{i \in \mathcal{X}} \tilde{\mathbf{a}}^T \tilde{\mathbf{K}}_i^T \mathbf{c}_i \mathbf{c}_i^T \tilde{\mathbf{K}}_i \tilde{\mathbf{a}} = \tilde{\mathbf{a}}^T \tilde{\mathbf{Q}}_{\mathcal{X}} \tilde{\mathbf{a}}$$

$$\text{where } \tilde{\mathbf{Q}}_{\mathcal{X}} = \sum_{i \in \mathcal{X}} \tilde{\mathbf{K}}_i^T \mathbf{c}_i \mathbf{c}_i^T \tilde{\mathbf{K}}_i.$$

Solving for the Model Parameters (cont.)

Since $\tilde{\mathbf{a}} = \begin{bmatrix} \mathbf{a} \\ 1 \end{bmatrix}$ and $\tilde{\mathbf{Q}}_{\mathcal{X}}$ is symmetric, we define $\tilde{\mathbf{Q}}_{\mathcal{X}} = \begin{bmatrix} \mathbf{Q} & \mathbf{q} \\ \mathbf{q}^T & q_0 \end{bmatrix}_{\mathcal{X}}$ so that

$$\mathcal{E}_{\mathcal{X}} = \tilde{\mathbf{a}}^T \tilde{\mathbf{Q}}_{\mathcal{X}} \tilde{\mathbf{a}} = \mathbf{a}^T \mathbf{Q} \mathbf{a} + 2 \mathbf{a}^T \mathbf{q} + q_0$$

$\mathcal{E}_{\mathcal{X}}$ is minimised when $\nabla_{\mathbf{a}} \mathcal{E}_{\mathcal{X}} = 2 \mathbf{Q} \mathbf{a} + 2 \mathbf{q} = \mathbf{0}$, so $\mathbf{a}_{\mathcal{X},\min} = -\mathbf{Q}^{-1} \mathbf{q}$.

The **choice of locality** \mathcal{X} will depend on application:

- If it is expected that the affine (or quadratic) model will apply accurately to the whole image, then \mathcal{X} can be the whole image (including all directions d and all selected scales s) and maximum robustness will be achieved.
- If not, then \mathcal{X} should be a smaller region, chosen to optimise the tradeoff between robustness and model accuracy. A good way to produce a smooth field is to make \mathcal{X} fairly small (e.g. a 32×32 pel region) and then to apply a smoothing filter across all the $\tilde{\mathbf{Q}}_{\mathcal{X}}$ matrices, element by element, before solving for $\mathbf{a}_{\mathcal{X},\min}$ in each region.

Constraint Weighting Factors C_i

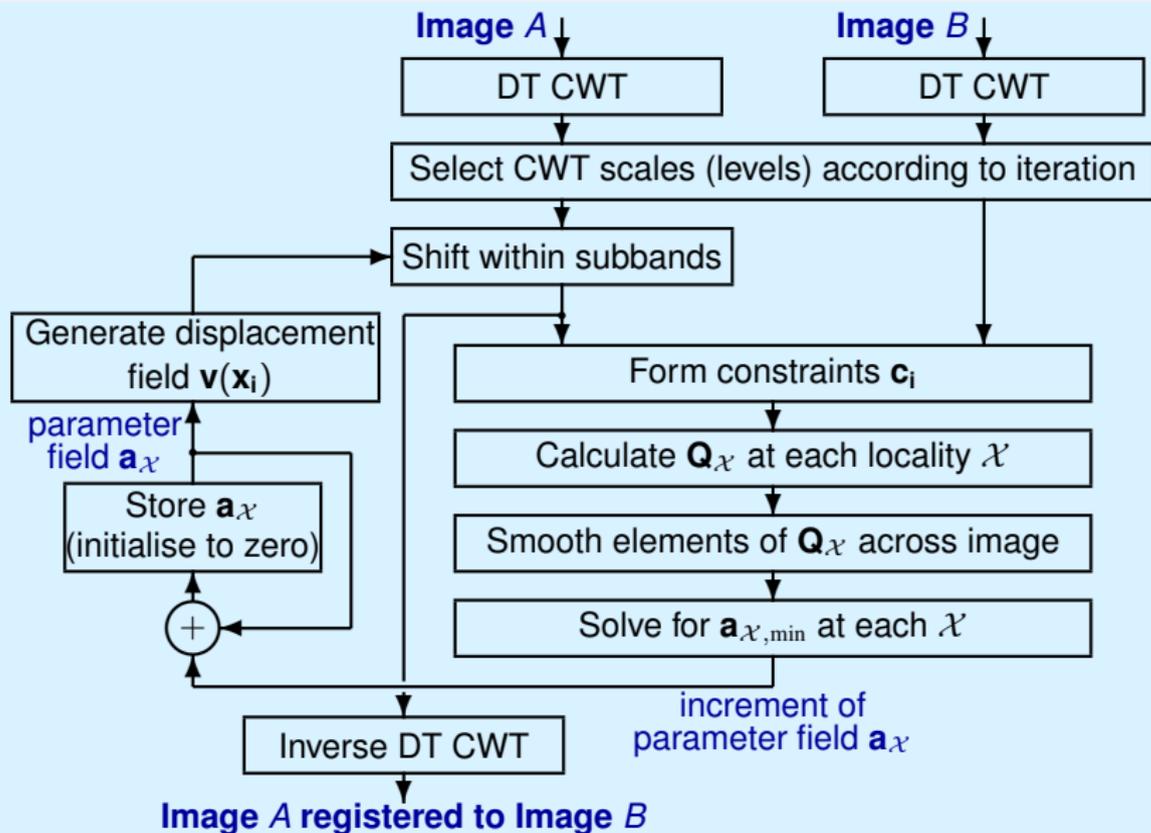
Returning to the equ. for the constraint vectors, $\mathbf{c}_i = C_i \begin{bmatrix} \nabla_{\mathbf{x}} \theta(\mathbf{x}_i) \\ \theta_B(\mathbf{x}_i) - \theta_A(\mathbf{x}_i) \end{bmatrix}$, the constant gain parameter C_i will determine how much weight is given to each constraint in $\tilde{\mathbf{Q}}_{\mathcal{X}} = \sum_{i \in \mathcal{X}} \tilde{\mathbf{K}}_i^T \mathbf{c}_i \mathbf{c}_i^T \tilde{\mathbf{K}}_i$.

Hemmendorf proposes some quite complicated heuristics for computing C_i , but for our work, we find the following gives **maximum weight to consistent sets of wavelet coefficients** and works well:

$$C_i = \frac{|d_{AB}|^2}{\sum_{k=1}^4 |u_k|^3 + |v_k|^3} \quad \text{where } d_{AB} = \sum_{k=1}^4 u_k^* v_k$$

and $\begin{bmatrix} u_1 & u_2 \\ u_3 & u_4 \end{bmatrix}$ and $\begin{bmatrix} v_1 & v_2 \\ v_3 & v_4 \end{bmatrix}$ are 2×2 blocks of wavelet coefficients centred on \mathbf{x}_i in images A and B respectively.

Registration Algorithm:



3-D Implications for the Phase-based Parametric Method

- \mathbf{x}_i and $\mathbf{v}(\mathbf{x}_i)$ become 3-element vectors, so \mathbf{c}_i and $\tilde{\mathbf{v}}_i$ become 4-vectors.
- For a 3-D affine model, \mathbf{K} becomes a 3×12 matrix, so that:

$$\mathbf{v}(\mathbf{x}) = \begin{bmatrix} 1 & 0 & 0 & x_1 & 0 & 0 & x_2 & 0 & 0 & x_3 & 0 & 0 \\ 0 & 1 & 0 & 0 & x_1 & 0 & 0 & x_2 & 0 & 0 & x_3 & 0 \\ 0 & 0 & 1 & 0 & 0 & x_1 & 0 & 0 & x_2 & 0 & 0 & x_3 \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ \vdots \\ a_{12} \end{bmatrix} = \mathbf{K}(\mathbf{x}) \cdot \mathbf{a}$$

and $\tilde{\mathbf{K}}$ becomes a 4×13 matrix.

- Hence $\tilde{\mathbf{Q}}_{\mathcal{X}} = \sum_{i \in \mathcal{X}} \tilde{\mathbf{K}}_i^T \mathbf{c}_i \mathbf{c}_i^T \tilde{\mathbf{K}}_i$ becomes a 13×13 symmetric matrix, containing $13 \times 7 = 91$ distinct elements per locality \mathcal{X} . At each selected scale s_i and spatial location \mathbf{x}_i in \mathcal{X} , there are now 28 subband directions d_j .

Demonstrations

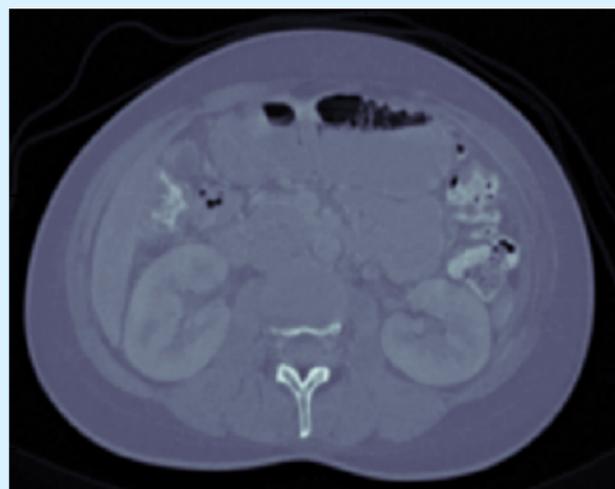
Enhancement of video corrupted by atmospheric turbulence

- 75 frames of video of a house on a distant hillside, taken through a high-zoom lens with significant turbulence of the intervening atmosphere due to rising hot air (courtesy of Don Fraser, ADFA).
- Task is to register each frame to a 'mean' image from the sequence, and then to reconstruct a high-quality still image from the registered sequence.

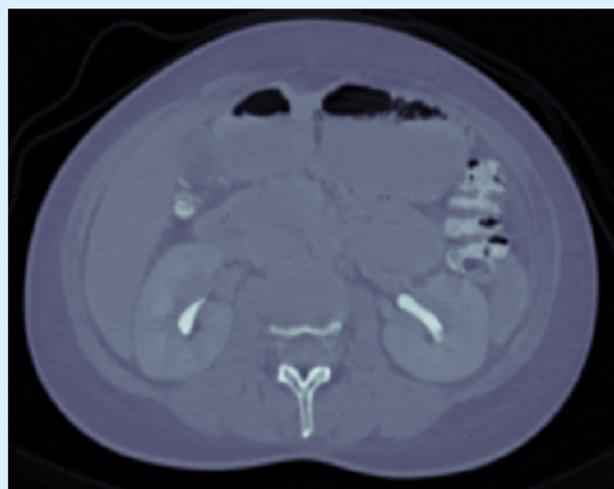
Registration of CT scans

- Two 3-D scans of the abdomen of the same patient, taken at different times with significant differences in position and contrast.
- Task is to register the two 3-D datasets as well as possible, despite the differences.

Tests of registration with 3D CT scans



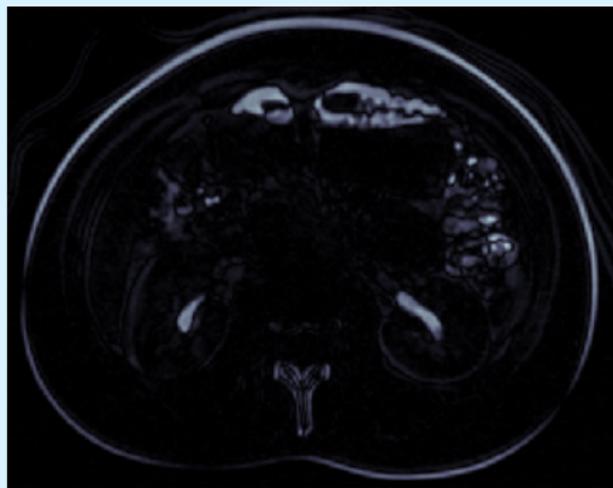
(a) Venous phase



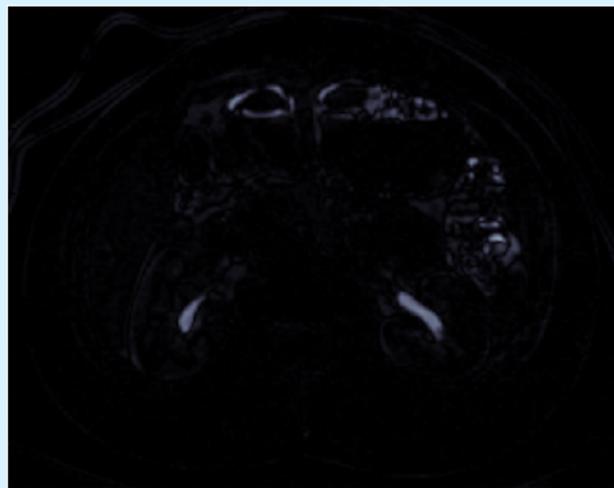
(b) Delay phase

Registration of 3-D CT scans ($384 \times 512 \times 128$) with contrast agent, where the venous phase dataset is being registered to the delay phase dataset. The registration is performed in 3-D, but only a single slice is displayed here for the convenience of visualization.

Tests of registration with 3D CT scans - difference images



(a) Before registration



(b) After registration

Difference between venous and delay phases, before and after registration.

Speed of registration

Our image registration algorithm is implemented in Matlab and tested with one 2.8 GHz CPU. The experimental datasets are of size $256 \times 256 \times 128$.

The simulation run times in seconds are:

- 3-D DT CWT on two input datasets: 5.8 s
- Iterations for affine parameter estimation:
 - 20.7 s for using level 5 coefficients
 - 23.1 s for using level 4 coefficients
 - 26.4 s for using level 3 coefficients
 - 25.2 s for using level 5 and level 4 coefficients
 - 28.2 s for using level 4 and level 3 coefficients
 - 66.4 s for using level 3 and level 2 coefficients
- 14.0 s to convert affine parameters to motion vectors
- 5.7 s to register the datasets by spatial shift

If 5 iterations are performed, the total time ≈ 150 s.

If level 2 is used in a 6th iteration, the total time ≈ 220 s.

Registration – conclusions

Our proposed algorithm for **robust registration** effectively combines

The Dual-Tree Complex Wavelet Transform

- Linear phase vs. shift behaviour
- Easy shiftability of subbands
- Directional filters select edge-like structures
- Good denoising of input images

with

Hemendorf's phase-based parametric method

- Finds LMS fit of parametric model to edges in images
- Allows simple filtering of \mathbf{Q}_x to fit more complex motions
- Integrates well with multiscale DT CWT structure

Application 2: Iterative Methods for Deconvolution

Bayesian Wavelet-based Deconvolution

Assume an image measurement process with blur \mathbf{H} and noise \mathbf{n} of variance σ_n^2 :

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}$$

Get **MAP estimate of \mathbf{x}** by minimising

$$J(\mathbf{x}) = \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 - \sigma_n^2 \log(p(\mathbf{x}))$$

where $p(\mathbf{x})$ represents the prior expectation about the image structure. It is often easiest to **model $p(\mathbf{x})$ in the wavelet domain**, with wavelet coeffs $\mathbf{w} = \mathbf{W}\mathbf{x}$ and $\mathbf{x} = \mathbf{M}\mathbf{w}$. Then we find \mathbf{w} to minimise

$$J(\mathbf{w}) = \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{M}\mathbf{w}\|^2 + \frac{1}{2} \mathbf{w}^T \mathbf{A}\mathbf{w}$$

where \mathbf{A} is diagonal and $A_{ii} = \sigma_n^2 / E(|w_i|^2)$, based on a **Gaussian Scale Mixture (GSM) model** for the wavelet coeffs w_i , $\forall i$ in vector \mathbf{w} .

Advantages of working with Wavelet Subbands

Simple steepest descent minimisation of $J(\mathbf{w})$ yields a gradient descent direction

$$-\nabla_{\mathbf{w}}J(\mathbf{w}) = \mathbf{M}^T \mathbf{H}^T (\mathbf{y} - \mathbf{H}\mathbf{M}\mathbf{w}) - \mathbf{A}\mathbf{w}$$

but this blurs the differences between \mathbf{y} and $\mathbf{H}\mathbf{M}\mathbf{w}$.

Subband emphasis can alleviate this and dramatically speed up convergence. We now minimise:

$$J(\mathbf{w}) = \frac{1}{2} \|\mathbf{y} - \mathbf{H} \underbrace{\sum_{j \in S} \mathbf{M}_j \mathbf{w}_j}_{\mathbf{x} = \mathbf{M}\mathbf{w}}\|^2 + \frac{1}{2} \sum_{j \in S} \mathbf{w}_j^T \mathbf{A}_j \mathbf{w}_j$$

where \mathbf{M}_j , \mathbf{A}_j and \mathbf{w}_j are *subband versions* of \mathbf{M} , \mathbf{A} and \mathbf{w} in which all entries apart from those in subband j have been set to zero.

The term $\|\mathbf{H}\mathbf{M}\mathbf{w}\|^2$ makes it difficult to minimise $J(\mathbf{w})$ because of all the *cross terms* in $\mathbf{w}^T \mathbf{M}^T \mathbf{H}^T \mathbf{H} \mathbf{M} \mathbf{w}$; so we use the ideas of Daubechies, Defrise & De Mol (2004) **on each subband independently**, as suggested by Vonesch & Unser (2008), to minimise $\bar{J}(\mathbf{w})$, an upper bound on $J(\mathbf{w})$.

Advantages of working with Wavelet Subbands (cont.)

Let

$$\bar{J}_n(\mathbf{w}) = J(\mathbf{w}) + \frac{1}{2} \sum_{j \in \mathcal{S}} \left(\alpha_j \|\mathbf{W}_j \mathbf{x}^{(n)} - \mathbf{w}_j\|^2 - \|\mathbf{H}\mathbf{M}_j(\mathbf{W}_j \mathbf{x}^{(n)} - \mathbf{w}_j)\|^2 \right)$$

where $\mathbf{x}^{(n)}$ is the estimate for \mathbf{x} at iteration n . As long as **each α_j is chosen to be no less than $|\mathbf{H}(\omega)|^2$ for all frequencies ω within the passband of subband j** , it can be shown that $\bar{J}_n(\mathbf{w}) \geq J(\mathbf{w})$, with approximate equality when \mathbf{w}_j is near $\mathbf{W}_j \mathbf{x}^{(n)}$.

The proof of this requires that the transform defined by \mathbf{W} and \mathbf{M} is a **tight frame** and that it is **shift invariant** so that $\mathbf{M}_j \mathbf{W}_j \mathbf{H} = \mathbf{H} \mathbf{M}_j \mathbf{W}_j$ – i.e. the transfer function of each subband can commute with the blurring function.

The Q-shift DT CWT approximately satisfies these criteria. The Shannon wavelet also satisfies these, but it is not compactly supported.

By choosing α_j optimally for each subband, we can overcome the problems of slow convergence of wavelet coefficients in spectral regions where \mathbf{H} has low gain.

The Resulting Algorithm:

$$\begin{aligned}
\bar{J}_n(\mathbf{w}) &= \frac{1}{2} \left(\|\mathbf{y} - \mathbf{H}\mathbf{M}\mathbf{w}\|^2 + \mathbf{w}^T \mathbf{A}\mathbf{w} \right. \\
&\quad \left. + \sum_{j \in \mathcal{S}} \alpha_j \|\mathbf{W}_j \mathbf{x}^{(n)} - \mathbf{w}_j\|^2 - \|\mathbf{H}(\mathbf{x}^{(n)} - \mathbf{M}\mathbf{w})\|^2 \right) \\
&= C(\mathbf{x}^{(n)}, \mathbf{y}) + \sum_{j \in \mathcal{S}} \left((\mathbf{H}\mathbf{x}^{(n)} - \mathbf{y})^T \mathbf{H}\mathbf{M}_j \mathbf{w}_j \right. \\
&\quad \left. + \frac{1}{2} \alpha_j \|\mathbf{W}_j \mathbf{x}^{(n)} - \mathbf{w}_j\|^2 + \frac{1}{2} \mathbf{w}_j^T \mathbf{A}_j \mathbf{w}_j \right)
\end{aligned}$$

where $C(\mathbf{x}^{(n)}, \mathbf{y})$ is independent of \mathbf{w} . This is a simple quadratic in \mathbf{w}_j , and its global minimum is achieved when $\partial \bar{J}_n(\mathbf{w}) / \partial \mathbf{w}_j = 0$. This gives

$$(\alpha_j \mathbf{I} + \mathbf{A}_j) \mathbf{w}_j = \alpha_j \mathbf{W}_j \mathbf{x}^{(n)} + \mathbf{M}_j^T \mathbf{H}^T (\mathbf{y} - \mathbf{H}\mathbf{x}^{(n)}) \quad \forall j$$

Hence, noting that $\mathbf{M}_j^T = \mathbf{W}_j$ for a tight frame, we get the new \mathbf{w}_j and \mathbf{x} :

$$\begin{aligned}
\mathbf{w}_j^{(n+1)} &= (\alpha_j \mathbf{I} + \mathbf{A}_j)^{-1} \left(\alpha_j \mathbf{W}_j \mathbf{x}^{(n)} + \mathbf{W}_j \mathbf{H}^T (\mathbf{y} - \mathbf{H}\mathbf{x}^{(n)}) \right) \quad \forall j \\
\mathbf{x}^{(n+1)} &= \mathbf{M} \sum_{j \in \mathcal{S}} \mathbf{w}_j^{(n+1)}
\end{aligned}$$

Updating the prior \mathbf{A}

Note: In the preceding analysis, we have assumed that all coefs in \mathbf{w} were purely real, and that complex transforms (like DT-CWT) created coefs whose real and imaginary parts were *separate real elements of \mathbf{w}* . However in the following, we assume that these parts have been combined together into *complex elements of \mathbf{w}* .

Bayesian analysis with a Gaussian scale mixture (GSM) model gives a diagonal prior matrix \mathbf{A} such that $A_{ii} = \sigma_n^2 / E(|w_i|^2)$.

In practise we use $A_{ii} = \frac{\sigma_n^2}{E(|w_i|^2) + \epsilon^2}$ so that

$$w_i^* A_{ii} w_i = \sigma_n^2 \frac{|w_i|^2}{E(|w_i|^2) + \epsilon^2} \approx \sigma_n^2 \|w_i\|_0$$

In this way we **maximise sparsity**, where ϵ defines the approximate threshold for $|w_i|$ between being *counted* or *not counted* in $\|w_i\|_0$. $E(|w_i|^2)$ is updated from the squared magnitudes of the complex coefs of $\mathbf{W}\mathbf{x}^{(n)}$ at each iteration n .

Updating the prior \mathbf{A} (cont.)

We call this function the L_{02} penalty, because

- It is closer to the L_0 -norm than to the L_1 -norm;
- It is smooth and differentiable (like the L_2 -norm) within each iteration of the algorithm.

But what are the expected wavelet variances, $E(|w_i|^2) \forall i$?

In practice, the estimated image is often contaminated by artifacts and noise, so the simple approach of calculating $E(|w_i|^2) = |w_i^{(n)}|^2$ direct from each complex coefficient in $\mathbf{W}\mathbf{x}^{(n)}$ does not work as well as we might hope.

We find we can obtain better estimates by calculating **denoised wavelet coefficients** $\hat{w}_i^{(n)}$ and setting $E(|w_i|^2) = |\hat{w}_i^{(n)}|^2$.

For denoising, we use the **Bayesian bi-variate shrinkage** algorithm of Sendur & Selesnick (2002), which models well the inter-scale (parent-child) dependencies of complex wavelet coefficients.

Initialisation and update strategies

- We initialise our algorithm with an under-regularised Wiener-like filter, implemented in the frequency domain:

$$\mathbf{x}^{(0)} = (\mathbf{H}^T \mathbf{H} + 10^{-3} \sigma_n^2 \mathbf{I})^{-1} \mathbf{H}^T \mathbf{y}$$

- Diagonal regularisation matrix \mathbf{A} is initialised using

$$A_{ii} = \frac{\sigma_n^2}{|\widehat{\mathbf{w}}_i|^2 + \epsilon^2} \quad \forall i, \quad \text{where } \widehat{\mathbf{w}} = \text{denoise}(\mathbf{W}\mathbf{x}^{(0)}) \quad \text{and } \epsilon = 0.01$$

- Optionally, \mathbf{A} is updated using $\widehat{\mathbf{w}} = \text{denoise}(\mathbf{W}\mathbf{x}^{(n)})$ at regular intervals in the iteration count n .

y : Cameraman, 9×9 uniform blur
+ noise at 40 dB PSNR



$x^{(0)}$: Initial image from
under-regularised Wiener-like filter



$\mathbf{x}^{(10)}$: Iteration 10 of DT CWT
with update of \mathbf{A}



$\mathbf{x}^{(0)}$: Initial image from
under-regularised Wiener-like filter



$\mathbf{x}^{(10)}$: Iteration 10 of DT CWT
with update of \mathbf{A}



$\mathbf{x}^{(30)}$: Iteration 30 of DT CWT
with update of \mathbf{A}



\mathbf{x} : Original
of Cameraman

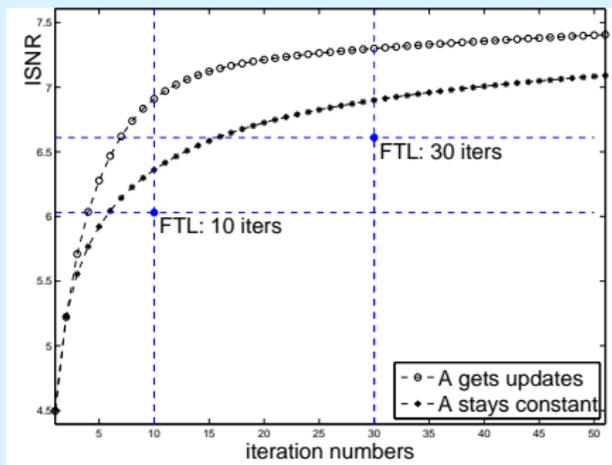


$\mathbf{x}^{(30)}$: Iteration 30 of DT CWT
with update of \mathbf{A}

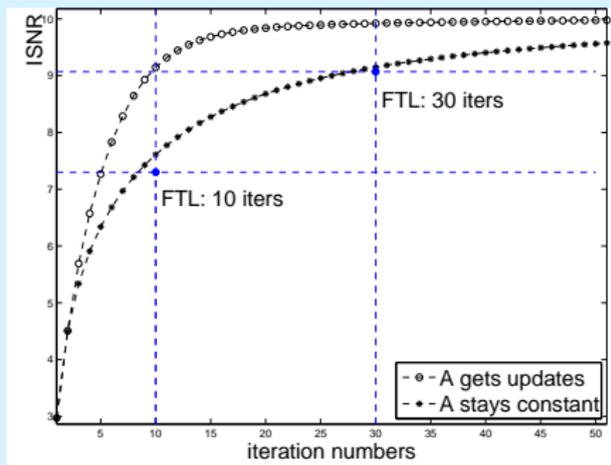


Convergence rate comparisons with Fast Thresholded Landweber algorithm (Vonesch & Unser)

Improvement in SNR (dB) of Cameraman image

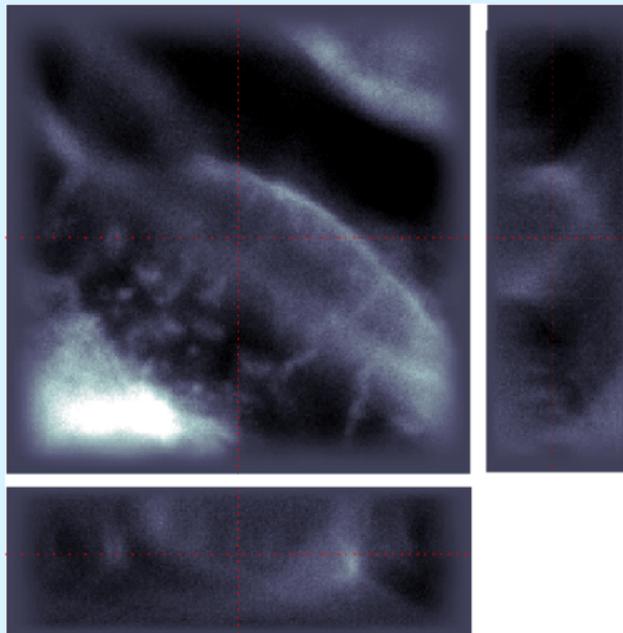


Improvement in SNR (dB) of House image

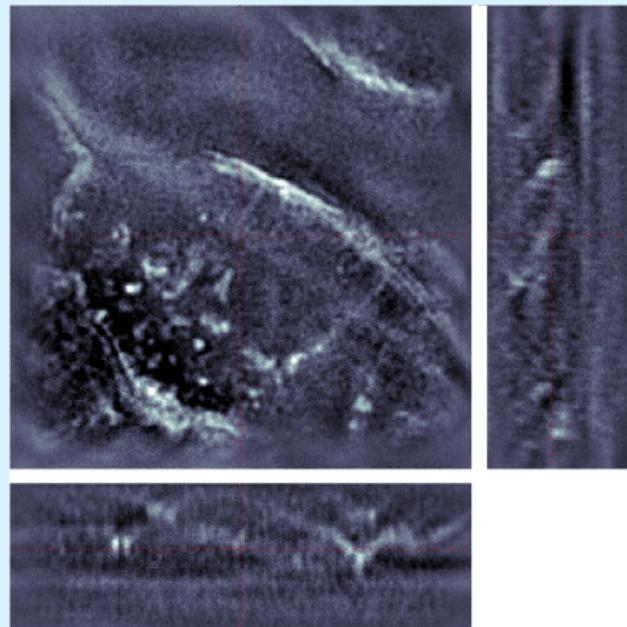


3D widefield fluorescence microscope data

y : 3D fluorescence data
with widefield imaging blur



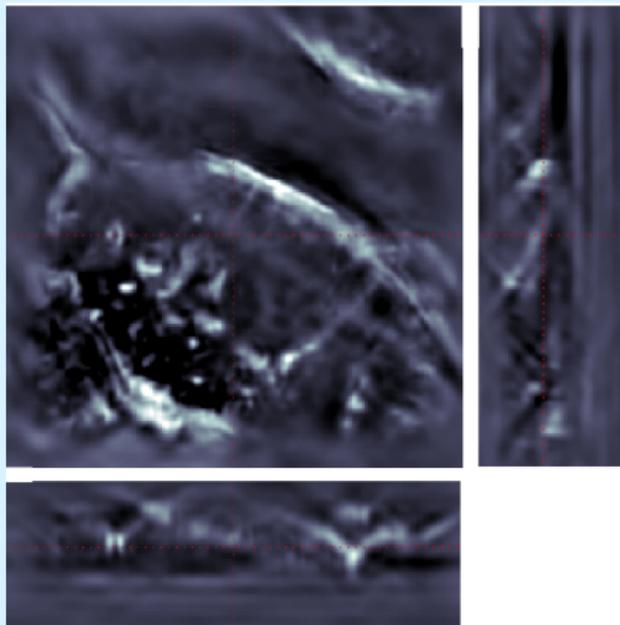
$x^{(0)}$: Initial data from
under-regularised Wiener-like filter



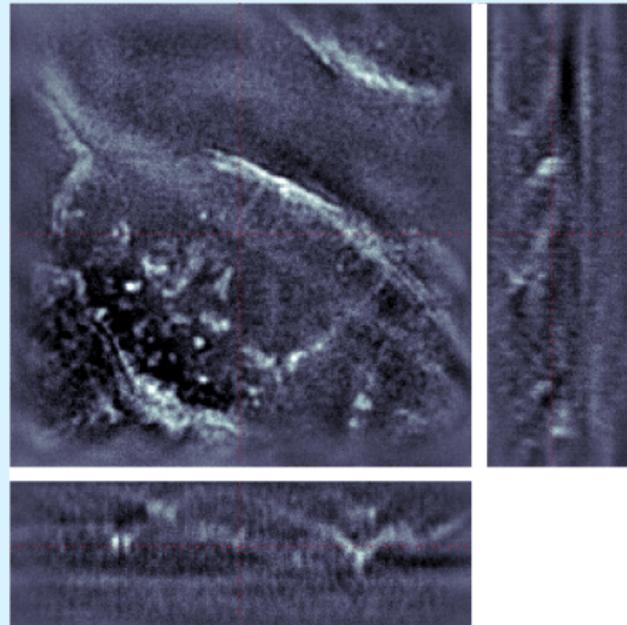
Size of 3D dataset = $256 \times 256 \times 80 = 5.24 \cdot 10^6$ voxels

3D widefield fluorescence microscope data

$\mathbf{x}^{(10)}$: Iteration 10 of DT CWT
with update of \mathbf{A}



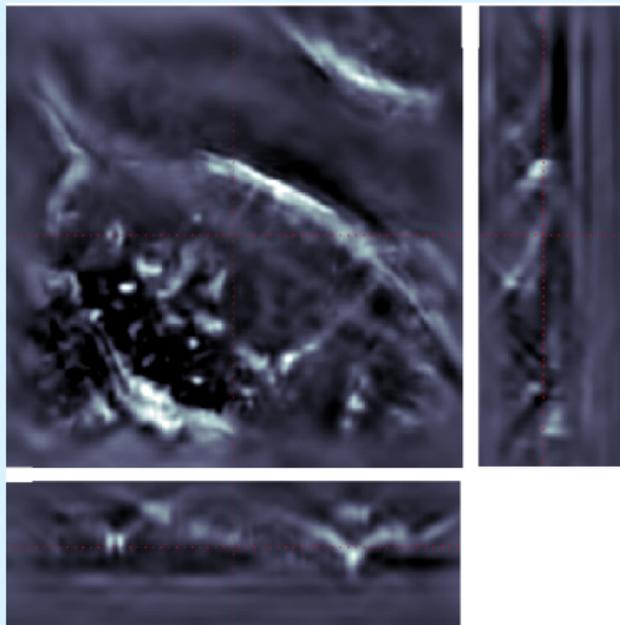
$\mathbf{x}^{(0)}$: Initial data from
under-regularised Wiener-like filter



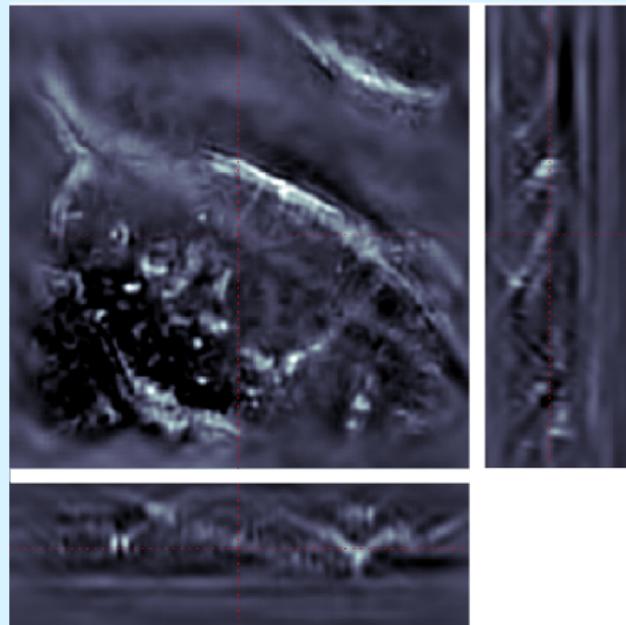
Size of 3D dataset = $256 \times 256 \times 80 = 5.24 \cdot 10^6$ voxels

3D widefield fluorescence microscope data

$\mathbf{x}^{(10)}$: Iteration 10 of DT CWT
with update of \mathbf{A}



$\mathbf{x}^{(30)}$: Iteration 30 of DT CWT
with update of \mathbf{A}



Size of 3D dataset = $256 \times 256 \times 80 = 5.24 \cdot 10^6$ voxels

Deconvolution – conclusions

- We have discussed some techniques for performing Deconvolution with overcomplete transforms, such as the DT \mathbb{C} WT.
- We have shown how sparsity helps with these types of large inverse problems, provided that the transform produces sparse representations of typical images (good directional selectivity helps here).
- We have discussed the reweighted L_2 penalty function and shown that Fast Thresholded Landweber (FTL) techniques may be used effectively with overcomplete transforms that possess tight-frame and shift-invariance properties, such as the DT \mathbb{C} WT.

Papers on complex wavelets and related topics are available at:

<http://www.eng.cam.ac.uk/~ngk/>