DUAL TREE
COMPLEX WAVELETS
Part 2

Nick Kingsbury

Signal Processing Group, Dept. of Engineering
University of Cambridge, Cambridge CB2 1PZ, UK.
ngk@eng.cam.ac.uk
www.eng.cam.ac.uk/~ngk

February 2005
Dual Tree Complex Wavelets

Part 1:

- Basic form of the DT CWT
- How it achieves shift invariance
- DT CWT in 2-D and 3-D – directional selectivity
- Application to image denoising

Part 2:

- Q-shift filter design
- How good is the shift invariant approximation
- Further applications – regularisation, registration, object recognition, watermarking.
Features of the Dual Tree Complex Wavelet Transform (DT CWT)

- Good **shift invariance**.

- Good **directional selectivity** in 2-D, 3-D etc.

- **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 1: 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. Figures in brackets indicate the approximate delay for each filter, where $q = \frac{1}{4}$ sample period.
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.

- Symmetric sub-sampling – see below.
**Q-shift DT CWT Basis Functions – Levels 1 to 3**

Figure 2: Basis functions for adjacent sampling points are shown dotted.
**Q-shift DT CWT Filter Design**

For the two trees we need lowpass filters with group delays which differ by **half a sample period**. This ensures low aliasing energy and hence good shift invariance.

The **Q-shift** version of the DT CWT achieves this with filters with group delays \( \approx \frac{1}{4} \) and \( \frac{3}{4} \) of a sample period, and has the following additional features:

- **Tree b** filters are the time-reverse of the **Tree a** filters.
- **Reconstruction** filters are the time-reverse of the **Analysis** filters.
- Bases are **orthonormal**, yielding a **tight-frame** transform.
- The complex bases are **linear phase**, since their magnitudes are symmetric and their phases are anti-symmetric (with a 45 degree offset).
Q-shift Filter Design Requirements

Fig. 2: 2-band analysis and reconstruction filter banks.

1. No aliasing: \[ G_1(z) = z H_0(-z) ; \quad H_1(z) = z^{-1} G_0(-z) \]

2. Perfect reconstruction: \[ H_0(z) G_0(z) + H_0(-z) G_0(-z) = 2 \]

3. Orthogonality: \[ G_0(z) = H_0(z^{-1}) \]

4. Group delay \( \simeq \frac{1}{4} \) sample period for \( H_0 \).

5. Good smoothness properties when iterated over scale.
FILTER DESIGN — DELAY

To get $2n$-tap lowpass filters, $H_0(z)$ and $G_0(z)$, with $\frac{1}{4}$ and $\frac{3}{4}$ sample delays:

- Design a $4n$-tap symmetric lowpass filter $H_{L2}(z)$ with half the required bandwidth and a delay of $\frac{1}{2}$ sample;

- **Subsample** $H_{L2}(z)$ by 2:1 to get $H_0(z)$ and $G_0(z)$.

![Impulse response of $H_{L2}(z)$ for $n = 6$. The $H_0$ and $G_0$ filter taps are shown as circles and crosses respectively.](image)

Fig. 3: Impulse response of $H_{L2}(z)$ for $n = 6$. The $H_0$ and $G_0$ filter taps are shown as circles and crosses respectively.
Filter Design – Perfect Reconstruction (PR)

For PR and orthogonality:

\[ H_0(z) G_0(z) = H_0(z) H_0(z^{-1}) \] must have no terms in \( z^{2k} \) except the term in \( z^0 \).

\[ \therefore H_0(z^2) H_0(z^{-2}) \] must have no terms in \( z^{4k} \) except the term in \( z^0 \).

But

\[ H_L2(z) = H_0(z^2) + z^{-1} H_0(z^{-2}) \]

and so

\[ H_L2(z) H_L2(z^{-1}) = 2 H_0(z^2) H_0(z^{-2}) + z^{-1} H_0^2(z^{-2}) + z H_0^2(z^2) \]

\[ \text{odd powers of } z \text{ only} \]

\[ \therefore H_L2(z) H_L2(z^{-1}) \] must have no terms in \( z^{4k} \) except the term in \( z^0 \).

Hence we can include PR as a direct design constraint on \( H_L2(z) H_L2(z^{-1}) \).
Filter Design — Smoothness

To obtain smoothness when iterated over many scales:

- **Ensure that the stopband of** $H_0(z)$ **suppresses energy at frequencies where unwanted passbands appear from subsampled filters operating at coarser scales.**

Consider the combined frequency response of $H_0$ over just two scales:

$$H_0(z) H_0(z^2)|_{z=e^{j\omega}} = H_0(e^{j\omega}) H_0(e^{2j\omega})$$

If the stopband of $H_0(e^{j\omega})$ covers $\omega_s \leq \omega \leq \pi$, then the unwanted transition band and passband of $H_0(e^{2j\omega})$ will extend from $\pi - \frac{\omega_s}{2}$ to $\pi$.

For $H_0(e^{j\omega})$ to suppress the unwanted bands of $H_0(e^{2j\omega})$ (see fig. 4):

$$\omega_s \leq \pi - \frac{\omega_s}{2} \quad \therefore \omega_s \leq \frac{2\pi}{3}$$
Fig. 4: Frequency responses of $H_{L2}(z)$ (blue), $H_0(z)$ (green), $H_0(z)H_0(z^2)$ (magenta), and the gain correction matrix $T$ (red) for $n = 6$ (12 taps for $H_0$).
Optimization for MSE in the Frequency Domain

We have now reduced the ideal design conditions for the length $4n$ symmetric lowpass filter $H_{L2}$ to be:

- Zero amplitude for all the terms of $H_{L2}(z)H_{L2}(z^{-1})$ in $z^{4k}$ except the term in $z^0$, which must be 1 (these are quadratic constraints on coef vector $h_{L2}$);

- Zero (or near-zero) amplitude of $H_{L2}(e^{j\omega})$ for the stopband, $\frac{\pi}{3} \leq \omega \leq \pi$ (these are linear constraints on $h_{L2}$).

If all constraints were linear, the LMS error solution for $h_{L2}$ could be found using a matrix pseudo-inverse method. \[ \because \text{we linearise the problem and iterate.} \]

If $h_{L2}$ at iteration $i$ is $h_i = h_{i-1} + \Delta h_i$, then

$$h_i * h_i = (h_{i-1} + \Delta h_i) * (h_{i-1} + \Delta h_i) = h_{i-1} * (h_{i-1} + 2\Delta h_i) + \Delta h_i * \Delta h_i$$

Since $\Delta h_i$ becomes small as $i$ increases, the final term can be neglected and the convolution ($*$) is expressed as a linear function of $\Delta h_i$. 
Hence we solve for $\Delta h_i$ such that:

$$\begin{align*}
C_{i-1} \left( h_{i-1} + 2\Delta h_i \right) &= [0 \ldots 0 1]^T \\
F \left( h_{i-1} + \Delta h_i \right) &\simeq [0 \ldots 0]^T
\end{align*}$$

where $C_{i-1}$ calculates every 4th term in the convolution with $h_{i-1}$, and $F$ evaluates the Fourier transform at $M$ discrete frequencies $\omega$ from $\frac{\pi}{3}$ to $\pi$ (typically $M \simeq 8n$)

Note that only one side of the symmetric convolution is needed in the rows of $C_{i-1}$, and the columns of $C_{i-1}$ and $F$ can be combined in pairs so that only the first half of the symmetric $\Delta h_i$ need be solved for.

To obtain **high accuracy solutions to the PR constraints**, we scale the equations in $C_{i-1}$ up by $\beta_i = 2^i$ to get the following iterative LMS method for $\Delta h_i$ and then $h_i$:

$$\begin{bmatrix} 2\beta_i C_{i-1} \\ F \end{bmatrix} \Delta h_i = \begin{bmatrix} \beta_i (c - C_{i-1} h_{i-1}) \\ -F h_{i-1} \end{bmatrix} \quad \text{with} \quad h_i = h_{i-1} + \Delta h_i$$

where $c = [0 \ldots 0 1]^T$. 
Two Final Refinements

- To include \textbf{transition band} effects, we scale rows of $F$ by diagonal matrix $T_i$, the gain (at iteration $i$) of $H_0(z^2)/H_0(1)$ at frequencies corresponding to $\frac{\pi}{3} \leq \omega \leq \frac{\pi}{2}$ in the frequency domain of $H_{L2}$ ($T_i$ is the red curve in fig. 4).

- To insert \textbf{predefined zeros} in $H_0(z)$ or $H_{L2}(z)$, we first note that a zero at $z = e^{j\pi}$ in $H_0$ will be produced by a pair of zeros at $z = e^{\pm j\pi/2}$ in $H_{L2}$. We can force zeros in $H_{L2}$ by forming a convolution matrix $H_f$ such that $H_f \ h'_i = h_i$, where $h'_i$ is the coef vector of the filter which represents all the zeros of $H_{L2}$ that are \textbf{not} predefined, and $H_f$ produces convolution with the predefined zeros.

Hence we now solve for $\Delta h'_i$ and then $h_i$ using

$$
\begin{bmatrix}
2\beta_i C_{i-1} \\
T_{i-1} F
\end{bmatrix} H_f \Delta h'_i = \begin{bmatrix}
\beta_i (c - C_{i-1} h_{i-1}) \\
-T_{i-1} F \ h_{i-1}
\end{bmatrix}
\quad \text{with} \quad h_i = h_{i-1} + H_f \Delta h'_i
$$
Fig. 5: Frequency responses of $H_{L2}(z)$ for $n = 8$ (blue), $n = 12$ (green) and $n = 16$ (red). Each filter has one predefined zero at $\omega = \frac{\pi}{2}$ and one at $\omega = \pi$. 
INITIALISATION

To initialise the iterative algorithm when $i = 1$, we must define $h_0$ and hence $C_0$ and $T_0$.

This is not critical and can be achieved by a simple inverse FFT of an ‘ideal’
lowpass frequency response for $H_{L2}(e^{j\omega})$ with a root-raised-cosine transition band covering the range

$$\frac{\pi}{6} < \omega < \frac{\pi}{3}$$

The impulse response is truncated symmetrically to length $4n$ to obtain $h_0$.

$C_0$ and $T_0$ may then be calculated from $h_0$.

CONVERGENCE

For some larger values of $n$, convergence can be slow. We have found this can be improved by using

$$h_i = h_{i-1} + \alpha H_f \Delta h'_i \quad \text{where} \quad 0 < \alpha < 1 \quad \text{(e.g. } \alpha \sim 0.8)$$
Results

• Figs. 4 and 5 show the frequency responses of $H_{L2}(z)$ for the cases $n = 6, 8, 12$ and 16, when there is one predefined zero at $\omega = \frac{\pi}{2}$ and one at $\omega = \pi$.

• Figs. 6 to 15 show, for a range of values of $n$, the impulse response of $H_{L2}(z)$, the level-4 DT CWT scaling functions and wavelets, the frequency responses of $H_0(z)$ and of $H_0(z) H_0(z^2)$, and the group delay of $H_0(z)$.

• Figs. 6 to 11 show these responses for the cases $n = 5, 6$ and 7, with either 0 or 1 predefined zero in $H_0(z)$ at $\omega = \pi$.

• Figs. 12 to 15 show these responses for the cases $n = 8, 12$ and 16, with 1 predefined zero in $H_0(z)$ at $\omega = \pi$.

Note how the responses improve with increasing $n$. The effect of predefining a zero in $H_0$ is in general quite small. More predefined zeros tend to degrade performance. $n = 7$ gives a good tradeoff between complexity and performance.
Fig. 6: Q-shift filters for $n = 5$ (10 filter taps) and no predefined zeros.
Fig. 7: Q-shift filters for $n = 5$ (10 filter taps) and 1 predefined zero at $\omega = \pi$. 
Fig. 8: Q-shift filters for $n = 6$ (12 filter taps) and no predefined zeros.
Fig. 9: Q-shift filters for $n = 6$ (12 filter taps) and 1 predefined zero at $\omega = \pi$. 
Fig. 10: Q-shift filters for $n = 7$ (14 filter taps) and no predefined zeros.
Fig. 11: Q-shift filters for $n = 7$ (14 filter taps) and 1 predefined zero at $\omega = \pi$. 
Fig. 12: Q-shift filters for $n = 8$ (16 filter taps) and 1 predefined zero at $\omega = \pi$. 
Fig. 13: Q-shift filters for $n = 10$ (20 filter taps) and 1 predefined zero at $\omega = \pi$. 
Fig. 14: Q-shift filters for $n = 12$ (24 filter taps) and 1 predefined zero at $\omega = \pi$. 
Fig. 15: Q-shift filters for $n = 16$ (32 filter taps) and 1 predefined zero at $\omega = \pi$. 
**Filter Design – Conclusions**

- The proposed algorithm gives a fast and effective way of designing Q-shift filters for the DT CWT.

- All filters produce perfect reconstruction, tight frames and linear-phase complex wavelets.

- As the length of the filters ($2n$) increases, the design method gives improvements in stopband attenuation, constancy of group delay, and smoothness in the resulting wavelet bases. Hence we get increasing accuracy of shift-invariance.

- The algorithm works well for filter lengths from 10 to over 50 taps.

- Matlab code for the algorithm and papers on the DT CWT can be downloaded from the author’s website, [http://www-sigproc.eng.cam.ac.uk/~ngk/](http://www-sigproc.eng.cam.ac.uk/~ngk/).

- Matlab code to implement the DT CWT is free for researchers and available by emailing the author at ngk@eng.cam.ac.uk.
**Visualising Shift Invariance**

- 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).

Fig 3 shows that the DT CWT has near-perfect shift invariance, whereas the maximally-decimated real discrete wavelet transform (DWT) has substantial shift dependence.
SHIFT INvariance OF DT CWT vs DWT

Figure 3: Wavelet and scaling function components at levels 1 to 4 of 16 shifted step responses of the DT CWT (a) and real DWT (b). If there is good shift invariance, all components at a given level should be similar in shape, as in (a).
Shift Invariance of simpler DT CWTs

Figure 4: Wavelet and scaling function components at levels 1 to 4 of 16 shifted step responses of simpler forms of the DT CWT, using (a) 14-tap and (b) 6-tap Q-shift filters with \( n = 7 \) and 5 respectively.
SHIFT INvariance – QUANTITATIVE MEASUREMENT

Basic configuration of the dual tree if either wavelet or scaling-function coefficients from just level $m$ are retained ($M = 2^m$).

Letting $W = e^{j2\pi/M}$, multi-rate analysis gives:

$$Y(z) = \frac{1}{M} \sum_{k=0}^{M-1} X(W^k z) [A(W^k z) C(z) + B(W^k z) D(z)]$$

For shift invariance, aliasing terms ($k \neq 0$) must be negligible. So we design $B(W^k z) D(z)$ to cancel $A(W^k z) C(z)$ for all non-zero $k$ that give overlap of the passbands of filters $C(z)$ or $D(z)$ with those of shifted filters $A(W^k z)$ or $B(W^k z)$. 

$$Y_a(z) \quad \downarrow M \quad \uparrow M \quad C(z) \quad Y(z)$$

$$Y_b(z) \quad \downarrow M \quad \uparrow M \quad D(z) \quad Y(z)$$

---

**Diagram:**

- **Tree a**
  - $X(z) \rightarrow A(z) \rightarrow \downarrow M \rightarrow \uparrow M \rightarrow C(z) \rightarrow Y_a(z)$
  - $B(z) \rightarrow \downarrow M \rightarrow \uparrow M \rightarrow D(z)$

- **Tree b**
  - $Y_a(z) \quad \downarrow M \quad \uparrow M \quad Y(z)$
  - $Y_b(z)$
A Measure of Shift Invariance

Since

\[ Y(z) = \frac{1}{M} \sum_{k=0}^{M-1} X(W^k z)[A(W^k z) C(z) + B(W^k z) D(z)] \]

we quantify the shift dependence of a transform by calculating the ratio of the total energy of the unwanted aliasing transfer functions (the terms with \( k \neq 0 \)) to the energy of the wanted transfer function (when \( k = 0 \)):

\[
R_a = \frac{\sum_{k=1}^{M-1} \mathcal{E}\{A(W^k z) C(z) + B(W^k z) D(z)\}}{\mathcal{E}\{A(z) C(z) + B(z) D(z)\}}
\]

where \( \mathcal{E}\{U(z)\} \) calculates the energy, \( \sum_r |u_r|^2 \), of the impulse response of a \( z \)-transfer function, \( U(z) = \sum_r u_r z^{-r} \).

\( \mathcal{E}\{U(z)\} \) may also be interpreted in the frequency domain as the integral of the squared magnitude of the frequency response, \( \frac{1}{2\pi} \int_{-\pi}^{\pi} |U(e^{j\theta})|^2 \, d\theta \) from Parseval’s theorem.
Types of DT CWT Filters

We show results for the following combinations of filters:

A. (13,19)-tap and (12,16)-tap near-orthogonal odd/even filter sets.

B. (13,19)-tap near-orthogonal filters at level 1, 18-tap Q-shift filters at levels \( \geq 2 \).

C. (13,19)-tap near-orthogonal filters at level 1, 14-tap Q-shift filters at levels \( \geq 2 \).

D. (9,7)-tap bi-orthogonal filters at level 1, 18-tap Q-shift filters at levels \( \geq 2 \).

E. (9,7)-tap bi-orthogonal filters at level 1, 14-tap Q-shift filters at levels \( \geq 2 \).

F. (9,7)-tap bi-orthogonal filters at level 1, 6-tap Q-shift filters at levels \( \geq 2 \).

G. (5,3)-tap bi-orthogonal filters at level 1, 6-tap Q-shift filters at levels \( \geq 2 \).
### Aliasing Energy Ratios,

Values of $R_a$ in dB, for filter types A to G over levels 1 to 5.

<table>
<thead>
<tr>
<th>Filters:</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>DWT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity:</td>
<td>2.0</td>
<td>2.3</td>
<td>2.0</td>
<td>1.9</td>
<td>1.6</td>
<td>1.0</td>
<td>0.7</td>
<td>1.0</td>
</tr>
<tr>
<td><strong>Wavelet</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Level 1</td>
<td>$-\infty$</td>
<td>$-\infty$</td>
<td>$-\infty$</td>
<td>$-\infty$</td>
<td>$-\infty$</td>
<td>$-\infty$</td>
<td>$-\infty$</td>
<td>-9.40</td>
</tr>
<tr>
<td>Level 3</td>
<td>-23.62</td>
<td>-27.93</td>
<td>-25.10</td>
<td>-20.32</td>
<td>-18.96</td>
<td>-14.60</td>
<td>-11.00</td>
<td>-3.53</td>
</tr>
<tr>
<td><strong>Scaling fn.</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Level 1</td>
<td>$-\infty$</td>
<td>$-\infty$</td>
<td>$-\infty$</td>
<td>$-\infty$</td>
<td>$-\infty$</td>
<td>$-\infty$</td>
<td>$-\infty$</td>
<td>-9.40</td>
</tr>
</tbody>
</table>
Application Examples

- **Regularisation** – e.g. for de-convolution, to avoid unwanted noise amplification.

- **Registration** – e.g. of panoramic images or motion of non-rigid bodies, such as medical images after time lapses.

- **Object recognition** – efficient searching for objects with known characteristics, without requiring precise location of the search template.

- **Watermarking** – making the watermark (noise) spectrum match the local properties of the host image.
Deconvolution Problem Formulation

Assume degradation of the image $\mathbf{x}$ is represented by a known stationary linear filter $H$ plus white noise $\mathbf{n}$ of zero mean and known variance $\sigma^2$.

In vector form for notational convenience, the degraded image $\mathbf{y}$ is given by:

$$\mathbf{y} = H\mathbf{x} + \mathbf{n}$$  \hfill (1)

For an image with $K$ pixels, $\mathbf{y}$, $\mathbf{x}$ and $\mathbf{n}$ will all be $K \times 1$ column vectors while $H$ will be a $K \times K$ (sparse) convolution matrix.

Note: Full matrix multiplications in this vector form are impractical since matrices would be very large (e.g. $K^2 = 256^4 \approx 4 \times 10^9$ elements for a typical $256 \times 256$ image), but other order-$K$ operations, such as transforms, convolutions and dot-products which we represent by matrix multiplications, are quite feasible.

For example the 2-D convolution, $H\mathbf{x}$ in (1) above, might be performed by a 2-D FFT, a dot-product (multiplication by a diagonal matrix) in the frequency domain, and then an inverse 2-D FFT.
Bayesian Deconvolution

For additive white Gaussian noise of variance $\sigma^2$, the likelihood of $y$, given $x$, is

$$p(y|x) = \prod_{i=1}^{K} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{([Hx]_i - y_i)^2}{2\sigma^2} \right\}$$

$$\propto \exp\left\{ -\frac{\|Hx - y\|^2}{2\sigma^2} \right\}$$

The MAP (maximum a posteriori) estimate of $x$ is then given by:

$$x_{MAP} = \arg\max_x p(x|y) = \arg\max_x p(y|x) p(x)$$

$$= \arg\min_x [-\log (p(y|x)) - \log (p(x))]$$

$$= \arg\min_x \left[ \frac{1}{2\sigma^2} \|Hx - y\|^2 + f(x) \right] \quad (2)$$

where $f(x) = -\log (p(x))$ – but what is this log expectation $f(x)$?
Bayesian Wavelet Deconvolution

Expectations about $x$ can most easily be formulated in the complex wavelet domain due to the transform’s good signal energy compaction properties and approximate shift invariance.

We represent the inverse DT CWT by a matrix $P$ such that $x = Pw$ is the image reconstructed from a vector of wavelet coefficients $w$.

We assume a scaled gaussian prior model for the complex wavelet coefficients (Re and Im parts), so that, following [Wang et al 1995], the prior pdf is given by $p(w) \propto \exp \{ -\frac{1}{2}w^TAw \}$ where $A$ is a diagonal matrix, such that $A_{ii}^{-1}$ is the expected variance of $w_i$ and $w^T$ is the complex-conjugate transpose of $w$.

Now $w_{MAP}$ (which produces $x_{MAP}$) is given by:

$$w_{MAP} = \arg\min_w [-\log(p(y|w)) - \log(p(w))]$$

$$= \arg\min_w \left[ \frac{1}{2\sigma^2} \|HPw - y\|^2 - \frac{1}{2}w^TAw \right] \quad (3)$$

Note that the variances in $A$ are allowed to vary between coefficients rather than being the same for all coefficients in a given subband.
Figure 5: Original *Cameraman* image (left) and version (right) blurred with a $9 \times 9$ uniform filter $H$ plus added white Gaussian noise of $\sigma = 0.555$ (BSNR = 40 dB).
**Energy Minimisation**

Our problem may now be formulated as:

Find the \( w \) which minimises the energy function

\[
E(w) = \frac{1}{2} \| HPw - y \|^2 + \frac{1}{2} w^T \sigma^2 A w
\]  

(4)

We attempt to minimise \( E(w) \) by repeating one-dimensional searches in sensible search directions. The steps in our method are:

1. **Estimate** \( P_x(f) \) the PSD of the image (e.g. Hillery and Chin method, 1991)

2. **Estimate the variances** of the noise and the wavelet coefficients to obtain \( \sigma^2 A \).

3. **Initialise the wavelet coefficients** to \( w^{(0)} \). Let \( k = 1 \).

4. **Calculate a search direction** \( h^{(k)} \) (using conjugate gradients).

5. **Minimise** \( E(w^{(k)}) \) along a line \( w^{(k)} = w^{(k-1)} + ah^{(k)} \). Update \( \hat{x}^{(k)} = Pw^{(k)} \).

6. **Repeat steps 4 and 5** for \( k = 2 \) to \( N \) (typically \( N \leq 20 \)).
**Conjugate Gradient Algorithm**

Differentiating equation (4) w.r.t. \( \mathbf{w} \):

\[
\nabla_\mathbf{w} E(\mathbf{w}) = P^T H^T (HP\mathbf{w} - \mathbf{y}) + \sigma^2 A\mathbf{w}
\]

Let \( \mathbf{g}^{(k)} = -\nabla_\mathbf{w} E(\mathbf{w}^{(k-1)}) \) be the steepest descent vector at iteration \( k \). Then, from Press et al. Numerical Recipes, the conjugate gradient vector is given by:

\[
\mathbf{h}^{(k)} = \mathbf{g}^{(k)} + \frac{|\mathbf{g}^{(k)}|^2}{|\mathbf{g}^{(k-1)}|^2} \mathbf{h}^{(k-1)} \quad \text{where} \quad \mathbf{h}^{(0)} = \mathbf{g}^{(0)}
\]

Since \( E \) is quadratic in \( \mathbf{w} \), the value of \( a \) which minimises \( E(\mathbf{w}^{(k)}) \) when \( \mathbf{w}^{(k)} = \mathbf{w}^{(k-1)} + a\mathbf{h} \) may be found analytically to be:

\[
a = \frac{-\mathbf{h}^T \nabla_\mathbf{w} E(\mathbf{w}^{(k-1)})}{\|HP\mathbf{h}\|^2 + \mathbf{h}^T \sigma^2 A\mathbf{h}} = \frac{\mathbf{h}^T \mathbf{g}^{(k)}}{\|HP\mathbf{h}\|^2 + \mathbf{h}^T \sigma^2 A\mathbf{h}}
\]

This requires no true matrix multiplications – \( \sigma^2 A \) is diagonal, \( P^T \) and \( P \) are forward and inverse CWTs, \( H \) and \( H^T \) are blurring convolutions (via FFT?).
Pre-Conditioning for Better Convergence

Conjugate Gradient descent converges most rapidly if the system is preconditioned such that its Hessian is a (scaled) identity matrix. BUT our matrices are much too large for this to be feasible (we need to invert the original Hessian)!

Instead we use a simple scaling of $w$ to produce a Hessian with diagonal entries of unity, but with (hopefully small) non-zero off-diagonal terms.

This preconditioning produces scaled wavelet coefficients $v = S^{-1}w$ where $S$ is diagonal. The Hessian of the energy in (4), as a function of $v$, is

$$\nabla^2_v E = S^TP^TH^THPS + S^T\sigma^2AS \tag{8}$$

The required scaling is $S_{ii} = 1/\sqrt{T_{ii}}$, where $T_{ii}$ is the $i^{th}$ diagonal entry of the Hessian $T = \nabla^2_w E = (P^TH^THP + \sigma^2A)$ of the original unscaled system.

The gradient in $v$-space is given by $g^{(k)} = -\nabla_v E = -S \nabla_w E$. 
CONJUGATE GRADIENT DECONVOLUTION BLOCK DIAGRAM

Figure 6:
Comparisons with other techniques

We have calculated the results of the DT-CWT and our version of standard Wiener and have listed them with results of others below (our results are in bold type). We see that the DT-CWT method gives the best performance. The WaRD method is shown to be 0.5 dB better than the multiscale Kalman filter, while the DT-CWT method is 0.7 to 1.0 dB better than the WaRD method (depending on number of iterations $N$).

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>ISNR /dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wiener (Banham and Katsaggelos)</td>
<td>3.58</td>
</tr>
<tr>
<td>Multiscale Kalman filter (B &amp; K)</td>
<td>6.68</td>
</tr>
<tr>
<td>Wiener (Neelamani et al)</td>
<td>5.37 (8.8 - 3.43)</td>
</tr>
<tr>
<td>Wiener (our version)</td>
<td>5.50</td>
</tr>
<tr>
<td>WaRD (Neelamani et al)</td>
<td>7.17 (10.6 - 3.43)</td>
</tr>
<tr>
<td>DT-CWT WaRD</td>
<td>7.05</td>
</tr>
<tr>
<td>DT-CWT CG, $N=10$</td>
<td>7.87</td>
</tr>
<tr>
<td>DT-CWT CG, $N=20$</td>
<td>8.13</td>
</tr>
<tr>
<td>DT-CWT CG, $N=50$</td>
<td>8.27</td>
</tr>
</tbody>
</table>
Results

The $256 \times 256$ Cameraman image with a uniform $9 \times 9$ blur and a blurred signal to noise ratio (BSNR) of 40dB has been used by [Banham and Katsaggelos 1996] and [Neelamani et al 1999], so we also use this setup to allow accurate comparisons with prior work. Deconvolving a uniform blur is difficult because of the large number of spectral zeros.

The DT-CWT used our standard (13,19)-tap near-orthogonal linear phase filters at level 1 and the 14-tap orthogonal Q-shift filters at levels $\geq 2$.

Figure 7 shows how our iterative Conjugate Gradient algorithm converges quite rapidly (within about 20 iterations) towards the maximum improvement in SNR of approx $1.2$ dB, while a simpler Steepest-Descent optimisation takes much longer to provide a similar improvement.
CONVERGENCE

Figure 7: Convergence of the conjugate gradient algorithm and a steepest descent version of the same algorithm.
Figure 8: Result of under-regularised Wiener filter (left) and wavelet denoised (WaRD) version of this (right). ISNR = 5.50 and 7.05 dB respectively.
Figure 9: Output images after iterations 1 (left) and 4 (right) of the Conjugate Gradient algorithm. ISNR = 7.22 and 7.57 dB respectively.
Figure 10: Output images after iterations 10 (left) and 50 (right) of the Conjugate Gradient algorithm. ISNR = 7.92 and 8.27 dB respectively.
APPLICATION EXAMPLES

- **Regularisation** – e.g. for de-convolution, to avoid unwanted noise amplification.

- **Registration** – e.g. of panoramic images or motion of non-rigid bodies, such as medical images after time lapses.

- **Object recognition** – efficient searching for objects with known characteristics, without requiring precise location of the search template.

- **Watermarking** – making the watermark (noise) spectrum match the local properties of the host image.
**Key Features of Robust Registration Algorithms**

- Edge-based methods are more robust than point-based ones.

- Must be automatic (no human picking of correspondence points) in order to achieve sub-pixel accuracy in noise.

- Bandlimited multiscale (wavelet) methods will allow spatially adaptive denoising.

- Phase-based bandpass methods can give rapid convergence and immunity to illumination changes between images.

- Displacement field should be smooth, so use of a wide-area parametric (affine) model is preferable to local translation-only models.
SELECTED METHOD

- Dual-tree Complex Wavelet Transform (DT CWT):
  - provides complex coefficients whose phase shift depends approximately linearly with displacement;
  - allows each subband of coefficients to be interpolated independently of other subbands (because of shift invariance).

- Parametric model of displacement field, whose solution is based on local edge-based motion constraints (Hemmendorf et al., IEEE Trans Medical Imaging, Dec 2002):
  - derives straight-line constraints from directional subbands of DT CWT;
  - solves for model parameters which minimise constraint error energy over multiple directions and scales.
**Parametric Model: Constraint equations**

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

A straight-line constraint on $v(x_i)$ can be written

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

For a phase-based system in which wavelet coefficients at $x_i$ in images $A$ and $B$ have phases $\theta_A$ and $\theta_B$, approximate phase linearity means that

$$c_i = C_i \begin{bmatrix} \nabla_x \theta(x_i) \\ \theta_B(x_i) - \theta_A(x_i) \end{bmatrix}$$

In practise we compute this by averaging finite differences at the centre of a $2 \times 2 \times 2$ block of coefficients from images $A$ and $B$.

$C_i$ is a constant which does not affect the line defined by the constraint, but which is important later.
PARAMETERS OF THE MODEL

We can define an affine parametric model 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

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

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

$$
\begin{bmatrix} \ldots & x_1 x_2 & 0 & x_1^2 & 0 & x_2^2 & 0 \\ \ldots & 0 & x_1 x_2 & 0 & x_1^2 & 0 & x_2^2 \end{bmatrix}
$$
SOLVING FOR THE MODEL PARAMETERS

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

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

$$c_i^T \tilde{v}_i = 0 \quad \text{when} \quad \tilde{v}_i = \tilde{K}_i \tilde{a} \quad \text{for all} \quad i \quad \text{such that} \quad x_i \in \mathcal{X}.$$

In practise this is an overdetermined set of equations, so we find the LMS solution, the value of $a$ which minimises the squared error

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

where $\tilde{Q}_\mathcal{X} = \sum_{i \in \mathcal{X}} (\tilde{K}_i^T c_i c_i^T \tilde{K}_i)$.
Solving for the Model Parameters (cont.)

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

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

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

The choice of locality $X$ will depend on application:

- If it is expected that the affine (or quadratic) model will apply accurately to the whole image, then $X$ can be the whole image and maximum robustness will be achieved.

- If not, then $X$ 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 $X$ fairly small (e.g. a $32 \times 32$ pel region) and then to apply a smoothing filter across all the $\tilde{Q}_X$ matrices, element by element, before solving for $a_{X,min}$ in each region.
**Constraint Weighting Factors**

Returning to the equation for the constraint vectors, \( c_i = C_i \begin{bmatrix} \nabla_x \theta(x_i) \\ \theta_B(x_i) - \theta_A(x_i) \end{bmatrix} \),

the constant gain parameter \( C_i \) will determine how much weight is given to each constraint in \( \tilde{Q}\chi = \sum_{i \in \chi} (\tilde{K}_i^T c_i c_i^T \tilde{K}_i) \).

Hemmendorf proposes some quite complicated heuristics for computing \( C_i \), but for the DT CWT, we find the following works well:

\[
C_i = \frac{|d_{AB}|^2}{\sum_{k=1}^4 |u_k|^3 + |v_k|^3}
\]

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 \( 2 \times 2 \) blocks of wavelet coefficients centred on \( x_i \) in images \( A \) and \( B \) respectively.
Image A

DT CWT

Select CWT levels according to iteration

Shift within subbands

Generate displacement field $v(x_i)$

parameter field $a_X$

Delay

Inverse DT CWT

Image A registered to image B

Image B

DT CWT

Form constraints $c_i$

Calculate $Q_X$ at each locality $X$

Smooth elements of $Q_X$ across image

Solve for $a_{X,\text{min}}$ at each $X$

increment of parameter field $a_X$
APPLICATION EXAMPLES

- **Regularisation** – e.g. for de-convolution, to avoid unwanted noise amplification.

- **Registration** – e.g. of panoramic images or motion of non-rigid bodies, such as medical images after time lapses.

- **Object recognition** – efficient searching for objects with known characteristics, without requiring precise location of the search template.

- **Watermarking** – making the watermark (noise) spectrum match the local properties of the host image.
Object recognition - using the Inter-Level Product (ILP)

Aim:

- To use the DT CWT to describe objects in images in ways that are relatively immune to moderate shifts (e.g. 4 to 8 pels in any direction) and yet preserve as much detail about the key object features as possible.

Problem:

- While CWT coef. magnitudes are immune to small shifts, their phases rotate quite rapidly with shift.
- CWT phases convey a lot of the information about the relative locations of key features.

Solution:

- Use the Inter-Level Product (ILP) to derotate the CWT phases at level $k$ using doubled phases of parent coefs. at level $k + 1$. (Matlab demo.)
APPLICATION EXAMPLES

- **Regularisation** – e.g. for de-convolution, to avoid unwanted noise amplification.

- **Registration** – e.g. of panoramic images or motion of non-rigid bodies, such as medical images after time lapses.

- **Object recognition** – efficient searching for objects with known characteristics, without requiring precise location of the search template.

- **Watermarking** – making the watermark (noise) spectrum match the local properties of the host image.
**Watermarking**

**Aim:**
- To minimise visibility of the watermark we must **match the local spectrum** of the pseudo-random watermark to the local spectrum of the host image.
- This allows the energy of the watermark to be maximised for a given (low) level of visibility and hence provides **maximum resilience** to attack.

**Method:**
- Apply the DT CWT separately to the host image and to the pseudo-random watermark (with flat spectrum). Use the magnitudes of the host image CWT coefs. to define the magnitudes of the watermark CWT coefs.
- Inverse CWT the scaled watermark coefs. to generate the spectrally matched watermark. This then forms a **spatially adaptive filter**.
- Combine this with the host – either using **addition** for basic spread spectrum modulation – or using **quantisation modulation** to minimise self-interference from the host. (**Matlab demo.**)
CONCLUSIONS

The Dual-Tree Complex Wavelet Transform provides:

- Approximate **shift invariance**
- **Directionally selective** filtering in 2 or more dimensions
- **Low redundancy** – only $2^m : 1$ for $m$-D signals
- **Perfect reconstruction**
- **Orthonormal filters** below level 1, but still giving **linear phase** (conjugate symmetric) complex wavelets
- **Low computation** – order-$N$; less than $2^m$ times that of the fully decimated DWT ($\sim 3.3$ times in 2-D, $\sim 5.1$ times in 3-D)
CONCLUSIONS (cont.)

• A general purpose multi-resolution front-end for many image analysis and reconstruction tasks:
  ◦ Enhancement (deconvolution)
  ◦ Denoising
  ◦ Motion / displacement estimation and compensation
  ◦ Texture analysis / synthesis
  ◦ Segmentation and classification
  ◦ Object recognition
  ◦ Watermarking
  ◦ 3D data enhancement and visualisation
  ◦ Coding (?)

Papers on complex wavelets are available at:
  http://www.eng.cam.ac.uk/~ngk/

A Matlab DTCWT toolbox is available on request from:
  ngk@eng.cam.ac.uk