Robust drift-free open-loop H.264 watermarking

Wei Chen Zafar Shahid Thomas Stütz
Florent Autrusseau Patrick Le Callet

June 13, 2013

This little code inset shows how the embedding is preformed. This is a simple Matlab (or GNU Octave) piece of code, showing how we can preserve the border coefficients of a 4×4 block in H.264.

```
%-- Initialize C and C^T:
c=[1 1 1 1; 1 0.5 -0.5 -1; 1 -1 -1 1; 0.5 -1 1 -0.5];
ct=c';

%-- Arbitrarily define an initial 4x4 block:
y=[1 2 3 4; 5 6 7 8; 9 10 11 12; 13 14 15 16];

%--Get the residual X:
x=ct*y*c

%-- Initialise the 2 solutions:
s1=[0 1 1 0; 0 1 1 0; 0 0 0 0; 0 0 0 0];
s2=[0 0 0 0; 1 1 0 0; 1 1 0 0; 0 0 0 0];

%-- Get DCTune's JND thresholds (jnd):
%-- ... we end up with 'jnd' (a 4x4 matrix).
%-- Out of the 4 non-zero coefficients of either s1 or s2,
%-- we get the minimum value.

jnd_s1 = min(min(jnd(1:2,2:3)));
jnd_s2 = min(min(jnd(2:3,1:2)));

%-- Apply equation 14 of the revised manuscript:
y1 = y + jnd_s1*s1;
x1=ct*y1*c

y2 = y + jnd_s2*s2;
x2=ct*y2*c

%-- Make sure the last row/column of x1 & x2 is unchanged:
diff_x1 = x - x1
diff_x2 = x - x2
```