Polyharmonic Local Cosine Transform for Improving the Reproduction Quality of JPEG-Compressed Images

Naoki Saito[1] *& Katsu Yamatani*

Department of Mathematics
University of California, Davis

May 16, 2006

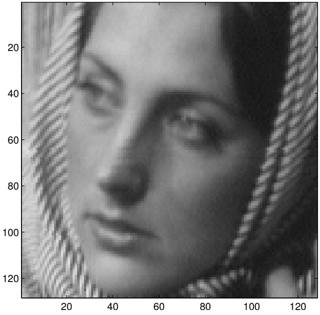
[1] Partially supported by grants from National Science Foundation and Office of Naval Research.

Outline

Outline

Motivations

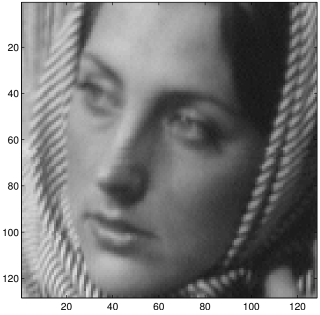
- Want to improve the quality of images (e.g., less blocking artifacts/visible discontinuities between blocks) reconstructed from the low bit rate JPEG files.



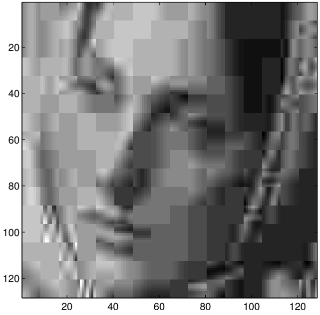
(a) Original: 8 bpp

Motivations

- Want to improve the quality of images (e.g., less blocking artifacts/visible discontinuities between blocks) reconstructed from the low bit rate JPEG files.



(a) Original: 8 bpp



(b) JPEG: 0.162 bpp

Motivations . . .

- Want to develop a local image transform that generates faster decaying expansion coefficients than block DCT used in JPEG and our previous Polyharmonic Local Sine Transform (PHLST) because the faster coefficient decay $\Longrightarrow$ more efficient compression
- Want to fully incorporate the infrastructure provided by the JPEG standard, e.g., the block DCT algorithm, the quantization method, the file format, etc.

Outline

Review of Fourier Cosine Series

- Let $\Omega = (0, 1)^2 \subset \mathbb{R}^2$ and $f \in C(\overline{\Omega})$ but not periodic: the periodically extended version of f is discontinuous at $\partial\Omega$.
- Then the size of the complex Fourier coefficients $c_{\mathbf{k}}$ of f decay as $O(\|\mathbf{k}\|^{-1})$, where $\mathbf{k} = (k_1, k_2) \in \mathbb{Z}^2$.
- Instead, expanding f into the Fourier cosine series gives rise to the decay rate $O(\|\mathbf{k}\|^{-2})$ because it is equivalent to the complex Fourier series expansion of the extended version of f via even reflection that is continuous at $\partial\Omega$.
- This is one of the main reasons why the JPEG Baseline method adopts Discrete Cosine Transform (DCT) instead of Discrete Fourier Transform (DFT).

Outline

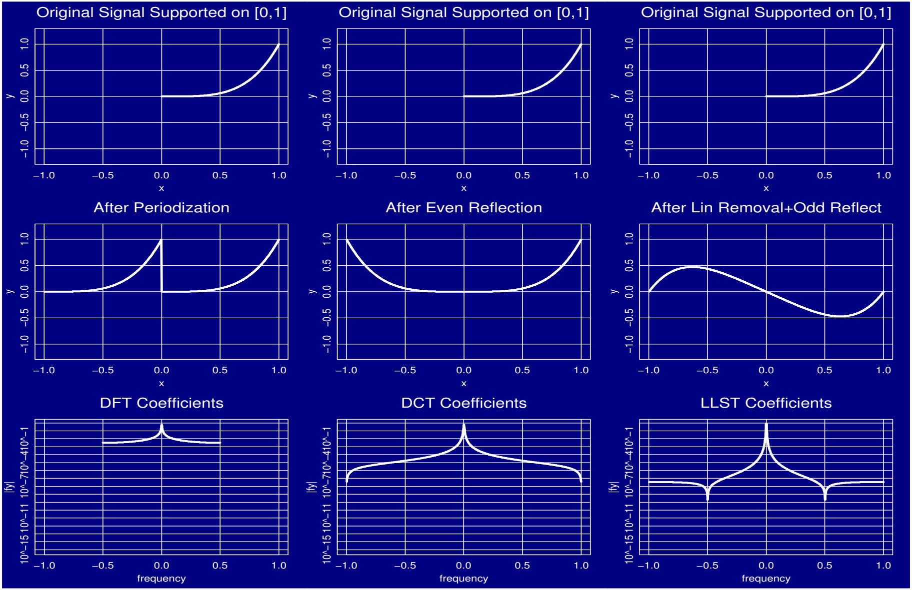
Review of Polyharmonic Local Sine Transform

- We now consider a decomposition $f = u + v$.
- The u (or polyharmonic) component satisfies Laplace's equation with the Dirichlet boundary condition.

$$\Delta u = 0 \quad \text{in } \Omega; \qquad u = f \quad \text{on } \partial\Omega.$$

- The u component is solely represented by the boundary values of f via the fast and highly accurate Dirichlet problem solver of Averbuch, Israeli, & Vozovoi (1998).
- The residual $v = f - u$ vanishes on $\partial\Omega \Longrightarrow$ The Fourier sine coefficients of v decay as $O(\|\mathbf{k}\|^{-3})$ for $v \in C^1(\overline{\Omega})$.
- See Saito & Remy (2003,2006) for the details.

Review of Polyharmonic Local Sine Transform . . .



Outline

Polyharmonic Local Cosine Transform

- Want to use DCT for fully utilizing the JPEG infrastructure.
- Want coefficients decaying faster than $O(\|\mathbf{k}\|^{-3})$.
- To do so, we need to solve Poisson's equation with the Neumann boundary condition:

$$\Delta u = K \quad \text{in } \Omega; \qquad \partial_\nu u = \partial_\nu f \quad \text{on } \partial\Omega,$$

  where the constant source term K (=the integration of $\partial_\nu f$ along $\partial\Omega$ normalized by the area of Ω) is necessary for solvability of the Neumann problem.
- Then, the Fourier cosine coefficients of the residual decay as $O(\|\mathbf{k}\|^{-4})$ for $v \in C^2(\overline{\Omega})$ because $\partial_\nu v = 0$ on $\partial\Omega$.

Why Poisson instead of Laplace?

Green's second identity claims that for any $u, v \in C^1(\overline{\Omega})$,

$$\int_\Omega (u\Delta v - v\Delta u)\,\mathrm{d}\mathbf{x} = \int_{\partial\Omega} (u\,\partial_\nu v - v\,\partial_\nu u)\,\mathrm{d}\sigma(\mathbf{x}),$$

where $\mathrm{d}\sigma(\mathbf{x})$ is a surface (or boundary) measure. Setting $v = 1$ with the Neumann boundary condition, we have

$$\int_\Omega \Delta u\,\mathrm{d}\mathbf{x} = \int_{\partial\Omega} \partial_\nu u\,\mathrm{d}\sigma(\mathbf{x}) = \int_{\partial\Omega} \partial_\nu f\,\mathrm{d}\sigma(\mathbf{x}).$$

This is a necessary condition that u must satisfy. Now, the source term of Poisson's equation is $K := \frac{1}{|\Omega|}\int_{\partial\Omega} \partial_\nu f\,\mathrm{d}\sigma(\mathbf{x})$, where $|\Omega|$ is the volume of the block Ω.

Outline

Outline

PHLCT from DCT coefficients

- Want to achieve the PHLCT representation of $f = u + v$ entirely in the DCT domain, $F = U + V$.
- Let $f(x, y) \in C(\overline{\Omega})$, and $f_{i,j}$ be a sample $f(x_i, y_j)$ with $x_i = (i + 0.5)/N$, $y_j = (j + 0.5)/N$, $i, j = 0, 1, \dots, N - 1$.
- Let $F \in \mathbb{R}^{N \times N}$ be a DCT coefficient matrix of $\{f_{i,j}\}$:

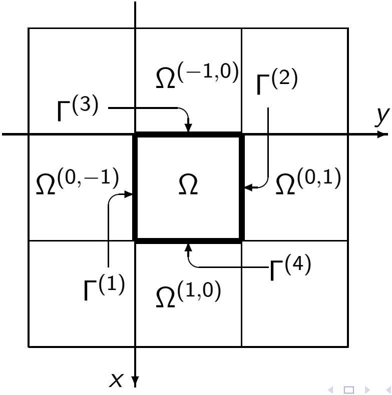
$$F_{k_1,k_2} := \lambda_{k_2} \sqrt{\frac{2}{N}} \sum_{j=0}^{N-1} \left(\lambda_{k_1} \sqrt{\frac{2}{N}} \sum_{i=0}^{N-1} f(x_i, y_j) \cos \pi k_1 x_i \right) \cos \pi k_2 y_j$$

  where $\lambda_0 = 1/\sqrt{2}$, $\lambda_k = 1$ for all $k \geq 1$.
- Now let's compute the DCT coefficient matrix U of the polyharmonic component u using F.

PHLCT from DCT coefficients . . .

- Assume for the moment that the discretized Neumann boundary data at each edge of $\overline{\Omega} = [0,1]^2$ are available:

$$g_i^{(1)} := -f_y(x_i, 0),\ g_i^{(2)} := f_y(x_i, 1),\ g_j^{(3)} := -f_x(0, y_j),\ g_j^{(4)} := f_x(1, y_j).$$



PHLCT from DCT coefficients . . .

- Let $\{G_k^{(\ell)}\}$ be the 1D-DCT coefficients of $\{g_i^{(\ell)}\}$.
- Then, we have a solution to Poisson's equation as (see Yamatani-Saito for details):

$$
\begin{aligned}
u(x,y) = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} \lambda_k \Big\{ & \Big(G_k^{(1)} \psi_k(y-1) + G_k^{(2)} \psi_k(y) \Big) \cos \pi kx \\
& + \Big(G_k^{(3)} \psi_k(x-1) + G_k^{(4)} \psi_k(x) \Big) \cos \pi ky \Big\} + c \,,
\end{aligned}
$$

where c is a constant to be determined and

$$
\psi_k(t) := \begin{cases} t^2/2 & \text{if } k = 0; \\ (\cosh \pi kt)/(\pi k \sinh \pi k) & \text{otherwise.} \end{cases}
$$

PHLCT from DCT coefficients . . .

- Applying 2D DCT to u above, we obtain $U = (U_{k_1,k_2})$ as

$$U_{k_1,k_2} = G_{k_1}^{(1)} \eta_{k_1,k_2} + G_{k_1}^{(2)} \eta_{k_1,k_2}^* + G_{k_2}^{(3)} \eta_{k_2,k_1} + G_{k_2}^{(4)} \eta_{k_2,k_1}^*,$$

where η_{k_1,k_2}, η_{k_1,k_2}^* are independent from the input image:

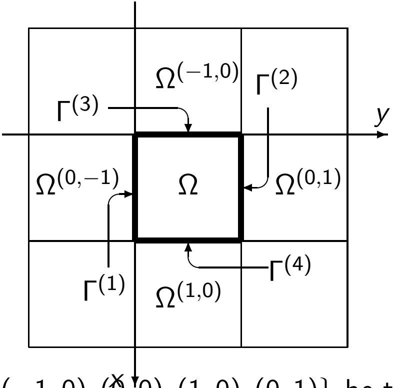
$$\begin{aligned} \eta_{k,m} &:= \lambda_m \sqrt{\frac{2}{N}} \sum_{i=0}^{N-1} \psi_k(x_i - 1) \cos \pi m x_i, \\ \eta_{k,m}^* &:= \lambda_m \sqrt{\frac{2}{N}} \sum_{i=0}^{N-1} \psi_k(x_i) \cos \pi m x_i, \end{aligned}$$

- Can set the DC component $U_{0,0} \equiv 0$ because the solution to the Poisson-Neumann problem is unique modulo an additive constant. In fact this is achieved by $c = -\frac{4N^2-1}{24N^{2.5}} \left(G_0^{(1)} + G_0^{(2)} + G_0^{(3)} + G_0^{(4)} \right)$. This will become important in our algorithms.

Outline

Approximation of the Neumann Boundary Data

- In practice, we need to estimate the Neumann boundary data $\{g_i^{(\ell)}\}$ from the image samples of the current and adjacent blocks. Let $f_{i,j}^{(s,t)} = f(x_i + s, y_j + t)$ and $\Omega^{(s,t)}$ be:



- Let $\mathcal{I}_5 := \{(0,-1), (-1,0), (0,0), (1,0), (0,1)\}$ be the indices of the current and adjacent blocks.

Approximation of the Neumann Boundary Data . . .

- Approximate $\{g_i^{(\ell)}\}$ using column & row averages:

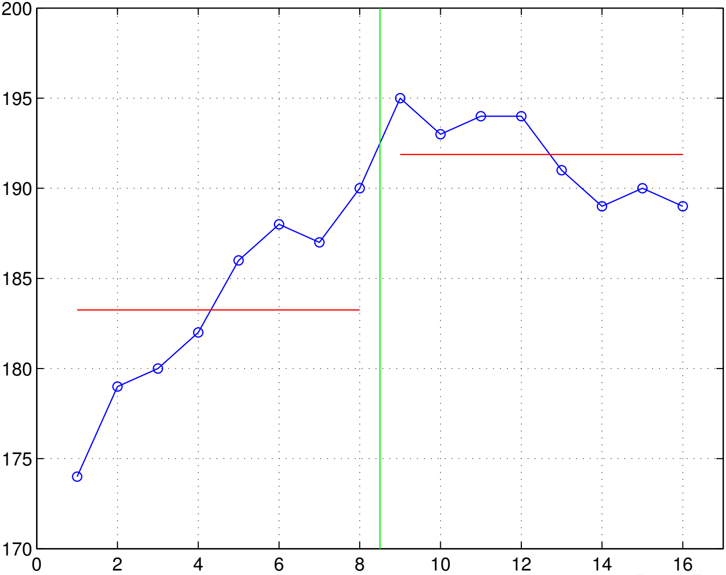
$$g_i^{(1)} \simeq X_i^{(-1)} - X_i^{(0)};\ g_i^{(2)} \simeq X_i^{(1)} - X_i^{(0)};\ g_j^{(3)} \simeq Y_j^{(-1)} - Y_j^{(0)};\ g_j^{(4)} \simeq Y_j^{(1)} - Y_j^{(0)}$$

$$X_i^{(t)} := \frac{1}{N}\sum_{j=0}^{N-1} f_{i,j}^{(0,t)},\quad Y_j^{(s)} := \frac{1}{N}\sum_{i=0}^{N-1} f_{i,j}^{(s,0)},$$

- Then, $\{G_k^{(\ell)}\}$ can be expressed using the first row & column of $F^{(s,t)}$. Consequently, for $(k_1, k_2) \neq (0,0)$, we have

$$U_{k_1,k_2} = \frac{1}{\sqrt{N}}\Big\{\left(F_{k_1,0}^{(0,-1)} - F_{k_1,0}\right)\eta_{k_1,k_2} + \left(F_{k_1,0}^{(0,1)} - F_{k_1,0}\right)\eta_{k_1,k_2}^{*} + \left(F_{0,k_2}^{(-1,0)} - F_{0,k_2}\right)\eta_{k_2,k_1} + \left(F_{0,k_2}^{(1,0)} - F_{0,k_2}\right)\eta_{k_2,k_1}^{*}\Big\}. \quad (1)$$

Approximation of the Neumann Boundary Data . . .



Outline

Modifying PHLCT for Practice

- Approximating Eq.(1) only using the DC components $F_{0,0}$ and $F_{0,0}^{(s,t)}$ allows us to simplify our algorithms: $U_{k_1,k_2} =$

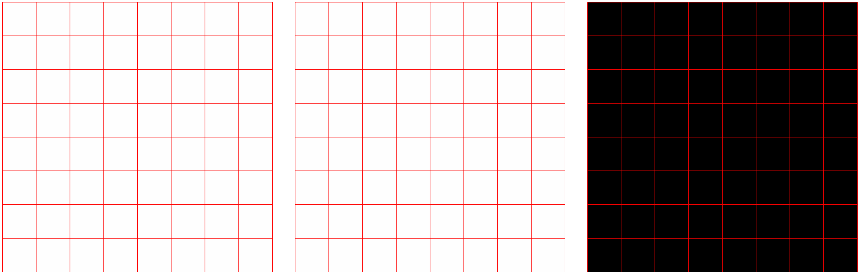
$$
(2) \begin{cases} 0 & \text{if } k_1 = k_2 = 0; \\ \frac{1}{\sqrt{N}} \left\{ \left(F_{0,0}^{(-1,0)} - F_{0,0} \right) \eta_{0,k_1} + \left(F_{0,0}^{(1,0)} - F_{0,0} \right) \eta_{0,k_1}^* \right\} & \text{if } k_1 \neq 0 = k_2; \\ \frac{1}{\sqrt{N}} \left\{ \left(F_{0,0}^{(0,-1)} - F_{0,0} \right) \eta_{0,k_2} + \left(F_{0,0}^{(0,1)} - F_{0,0} \right) \eta_{0,k_2}^* \right\} & \text{if } k_1 = 0 \neq k_2; \\ U_{k_1,k_2} \text{ as Eq.(1)} & \text{otherwise.} \end{cases}
$$

- Now set $V_{k_1,k_2} = F_{k_1,k_2} - U_{k_1,k_2}$, $\forall k_1, k_2$. Note $V_{0,0} = F_{0,0}$!!
- Note also that if we know V of the current and adjacent blocks, we can reconstruct F. No need to store U! See next page.
- Strictly speaking, this new version of u does not satisfy Poisson's equation, but satisfies the Neumann condition.

Inverse PHLCT

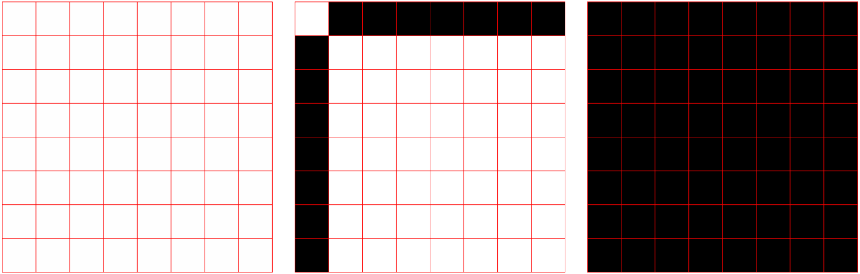
1. Assuming $V^{(s,t)}$, $(s,t) \in \mathcal{I}_5$, are available, recover the first column and row of U using the DC components, $F_{0,0}^{(s,t)} \left(= V_{0,0}^{(s,t)}\right)$, $(s,t) \in \mathcal{I}_5$ via (2);
2. Recover the first column and row of $F^{(s,t)}$, $(s,t) \in \mathcal{I}_5$ by summing those of U and V (see (2));
3. Recover other entries of U via (1) and the results of Step 2;
4. Set $F = U + V$;
5. Apply Inverse 2D DCT to F to recover f.

Inverse PHLCT: Step 0



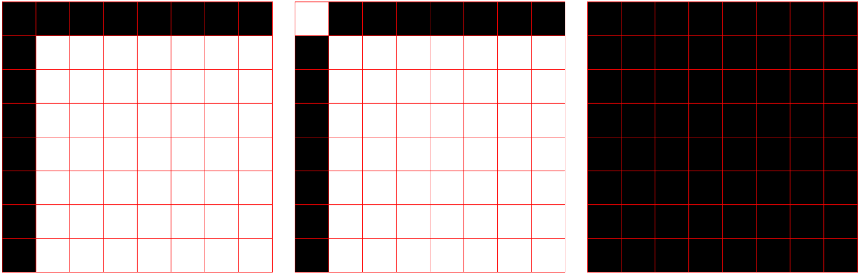
(a) F (b) U (c) V

Inverse PHLCT: Step 1



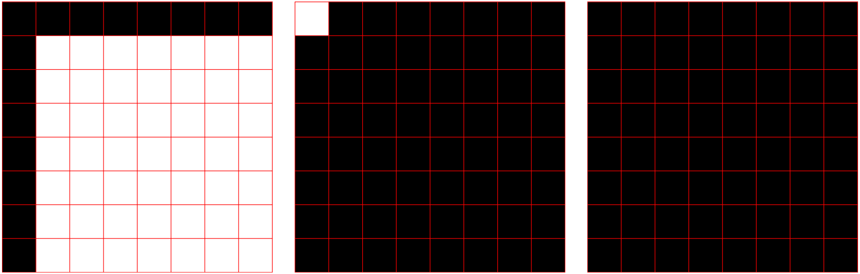
(a) F (b) U (c) V

Inverse PHLCT: Step 2



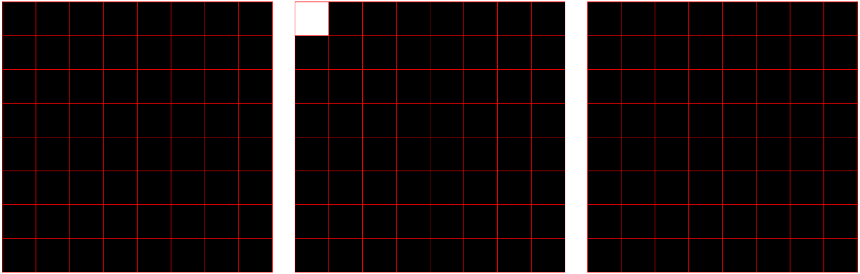
(a) F (b) U (c) V

Inverse PHLCT: Step 3



(a) F (b) U (c) V

Inverse PHLCT: Step 4



(a) F (b) U (c) V

Outline

Full Mode PHLCT (FPHLCT)

- FPHLCT adds simple procedures in both the encoder and the decoder parts of the JPEG Baseline method.
- In the encoder part, the only difference from JPEG is to: 1) compute U from F; and 2) compute the residual $V = F - U$ and store the quantized version V^Q instead of F^Q.
- In the decoder part, the only difference from JPEG is to: 1) compute U^Q, the estimate of U from V^Q; and 2) compute $U^Q + V^Q$ as an improved estimate of F over F^Q.
- Because V decays faster than F, the decompressed image quality gets better than JPEG if it is compressed at the same bit rate.

Outline

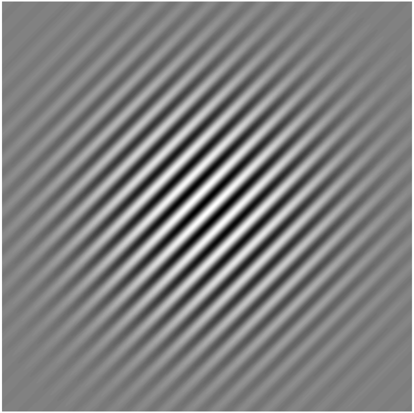
Partial Mode PHLCT (PPHLCT)

- Only the decoder part of the JPEG Baseline method is modified: PPHLCT accepts the JPEG-compressed files.
- The JPEG encoder kills small DCT coefficients $F_{\mathbf{k}'}$, i.e., $F^Q_{\mathbf{k}'} = 0$.
- PPHLCT replaces those $F^Q_{\mathbf{k}'}$ by $U^Q_{\mathbf{k}'}$ if $U^Q_{\mathbf{k}'}$ are also small.
- This is possible because U^Q can be computed solely from the first column & row of F^Q and those of the adjacent blocks $F^{(s,t)Q}$; see Eqs.(1), (2).
- Our reasoning to do this is $F_{\mathbf{k}} \approx U_{\mathbf{k}}$ for large $\mathbf{k}$ because $V_{\mathbf{k}}$ decays quickly.
- We also add some quadratic polynomial to reduce the blocking artifacts further. This can be done also in the DCT domain. (See Yamatani-Saito for the details.)

Outline

Numerical Experiments

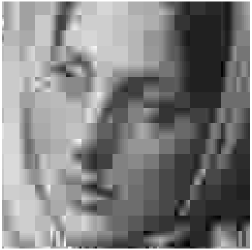
(a) Barbara



(b) Gabor

Two test images.

Numerical Experiments . . .



(a) JPEG, 23.61 dB

(b) FPHLCT, 24.19 dB

(c) PPHLCT, 23.97 dB

Compressed at 0.15 bits/pixel. Numerical values indicate the Peak Signal-to-Noise Ratio (PSNR).

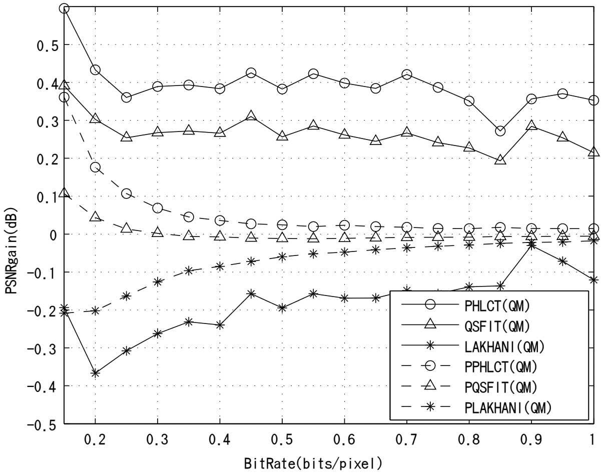
(a) JPEG, 25.67 dB

(b) FPHLCT, 26.05 dB

(c) PPHLCT, 25.73 dB

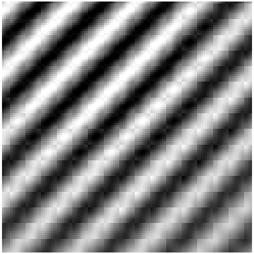
Compressed at 0.30 bits/pixel. Numerical values indicate the Peak Signal-to-Noise Ratio (PSNR).

Numerical Experiments . . .

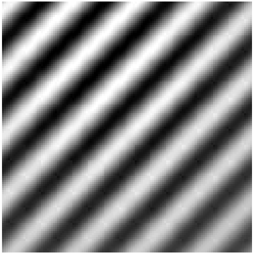


Comparison of PSNR gain by various methods for the Barbara image over the JPEG Baseline method.

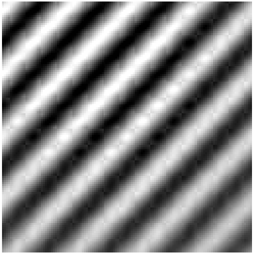
Numerical Experiments . . .



(a) JPEG, 31.41 dB

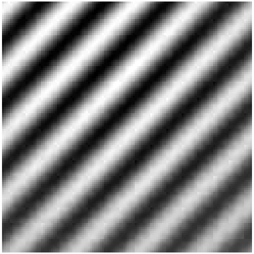


(b) FPHLCT, 39.21 dB

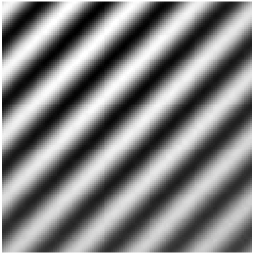


(c) PPHLCT, 35.69 dB

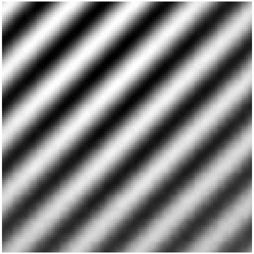
Compressed at 0.15 bits/pixel. Numerical values indicate the Peak Signal-to-Noise Ratio (PSNR).



(a) JPEG, 38.12 dB



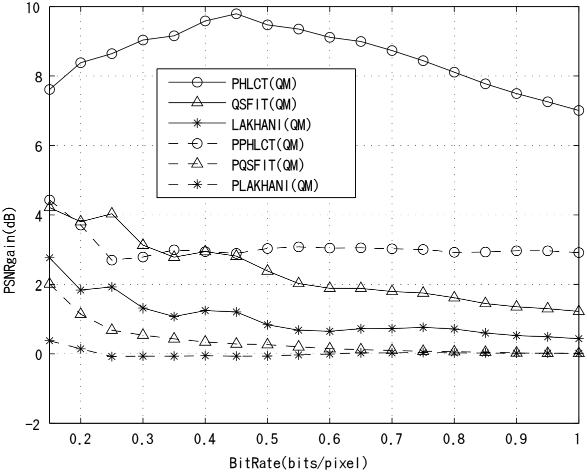
(b) FPHLCT, 47.02 dB



(c) PPHLCT, 40.89 dB

Compressed at 0.30 bits/pixel. Numerical values indicate the Peak Signal-to-Noise Ratio (PSNR).

Numerical Experiments . . .



Comparison of PSNR gain by various methods for the Gabor image over the JPEG Baseline method.

Outline

Conclusion

- More extensive numerical experiments (see Yamatani-Saito) indicate that FPHLCT reduces the bit rates about 15% over JPEG whereas PPHLCT does about 7% to achieve the same PSNR in the relatively low bit rate range.
- If one can afford to use the higher bit rates, then our methods naturally approach to the performance of JPEG.
- PPHLCT is particularly useful because it accepts the files compressed by the JPEG standard.
- On the other hand, FPHLCT is better than PPHLCT if one can afford to modify the encoder part of the JPEG standard.
- Additional computational cost of both methods over JPEG is small: linearly proportional to the number of pixels of an input image.
- Should be useful for zooming, interpolation, feature extraction.

Outline

References

- K. Yamatani & N. Saito: "Improvement of DCT-based compression algorithms using Poisson's equation," to appear in *IEEE Trans. Image Proc.*, 2006.
- N. Saito & J.-F. Remy: "The polyharmonic local sine transform: A new tool for local image analysis and synthesis without edge effect," *Appl. Comp. Harm. Anal.*, vol.20, no.1, pp.41–73, 2006.
- A. Averbuch, M. Israeli, & L. Vozovoi: "A fast Poisson solver of arbitrary order accuracy in rectangular regions," *SIAM J. Sci. Comp.*, vol.19, no.3, pp.933–952, 1998.
- G. K. Wallace: "The JPEG still picture compression standard," *IEEE Trans. Consumer Electronics*, vol.38, no.1., pp.xviii–xxxiv, 1992.

Thank you very much for your attention!