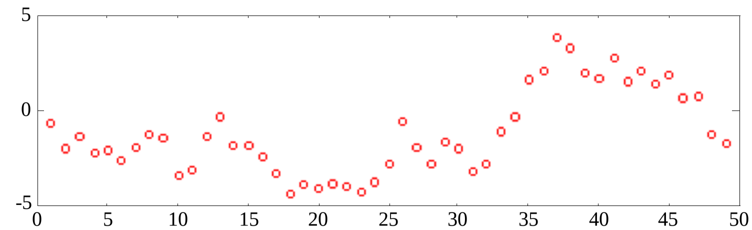
DPCM (DIFFERENTIAL PULSE CODE MODULATION)

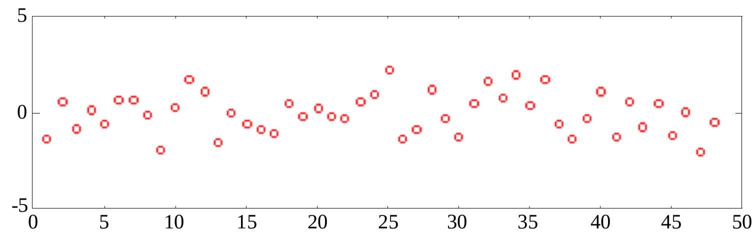
Goal: An effective low complexity quantizer for correlated sources based on scalar quantization.

Consider quantizing a highly correlated, i.e. slowly varying, source, e.g. AR Gauss, with $\rho = .95$, $EX = 0$, $\sigma^2 = 3.2$. A typical sample function $\Rightarrow$



For future comparison, optimal scalar quantization with R = 2 yields D = 0.37.

Notice: Most samples are similar to their predecessors. Successive sample differences, as shown to the right $\Rightarrow$, are Gaussian with mean zero and very small variance.



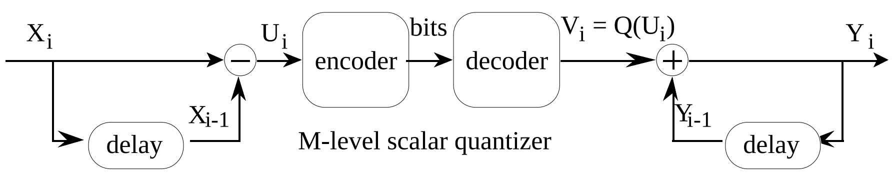
$$E(X_i-X_{i-1})^2 = EX_i^2 - 2\,EX_iX_{i-1} + EX_{i-1}^2 = 2\sigma^2 - 2\rho\sigma^2 = 0.1\sigma^2 = 0.32 << \sigma^2 = 3.2$$

This suggests quantizing successive sample differences.

In comparison, to direct scalar quantization of X, scalar quantization of (X_i-X_{i-1}) with R = 2 yields $D = 0.37 \times 0.32/3.2 = 0.037 << 0.37$.

CODING SAMPLE DIFFERENCES

ATTEMPT 1: NAIVE DIFFERENTIAL CODING



- Assume $X_0 = Y_0 = 0$
- Rate = rate of scalar quantizer.
- Key equation: $Y_i = Y_{i-1} + Q(X_i-X_{i-1})$
- Consider the errors in the reproductions

$$X_1 - Y_1 = X_1 - Q(X_1)$$

$$X_2 - Y_2 = X_1+U_2 - (Y_1+V_2) = (X_1-Y_1) + (U_2-V_2)$$

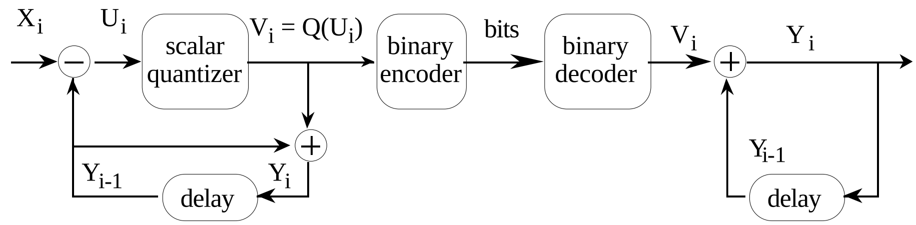
$$X_3 - Y_3 = X_2+U_3 - (Y_2+V_3) = (X_2-Y_2) + (U_3-V_3) = (X_1-Y_1) + (U_2-V_2) + (U_3-V_3)$$

$$X_i - Y_i = X_{i-1}+U_i - (Y_{i-1}+V_i) = (X_{i-1}-Y_{i-1}) + (U_i-V_i) = (X_1-Y_1) + (U_2-V_2) + (U_3-V_3) + \ldots + (U_i-V_i)$$

- Errors accumulate. This method does not work.

ATTEMPT 2: DIFFERENTIAL PULSE CODE MODULATION (DPCM)

Traditional block diagram:



Assume $Y_0 = 0$. Decoder is same as before.

Key Equations:

- $Y_i = Y_{i-1} + Q(X_i - Y_{i-1})$ i.e. new reproduction equals old reproduction plus quantized prediction error
- $X_i - Y_i = U_i - Q(U_i)$ i.e. overall error = error introduced by scalar quantizer $\left(X_i - Y_i = (Y_{i-1}+U_i) - (Y_{i-1}+Q(U_i)) = U_i - Q(U_i)\right)$

Notes:

- Rate = rate of scalar quantizer.
- $U_i \neq X_i - X_{i-1}$ (For naive encoder $U_i = X_i - X_{i-1}$). However, $U_i = X_i - Y_{i-1} \cong X_i - X_{i-1}$. This implies $\sigma_U^2 << \sigma_X^2$, so quantizing U gives less MSE than quantizing X. Since $X_i - Y_i = U_i - Q(U_i)$, the small MSE for U carries over to X.
- Therefore DPCM works well.
- The predictive/differential coding style in DPCM also appears in other lossy coding methods, e.g. interframe video coding.

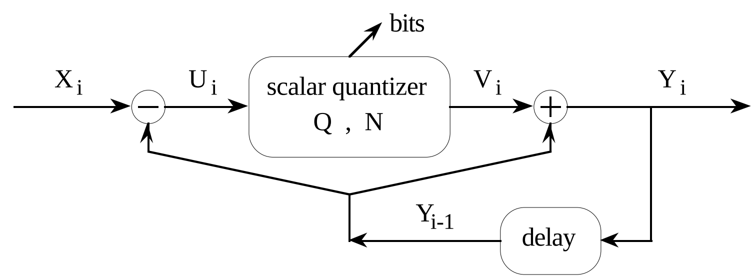
DPCM: ADAPTIVE QUANTIZATION VIEWPOINT

Recall: $Y_i = Y_{i-1} + Q(X_i - Y_{i-1})$

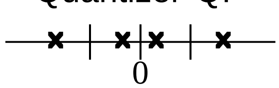
$\Rightarrow$ X_i is quantized with shifted quantizer :

$Y_i = b + Q(X_i - b)$,

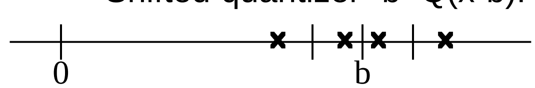
where $b = Y_{i-1}$

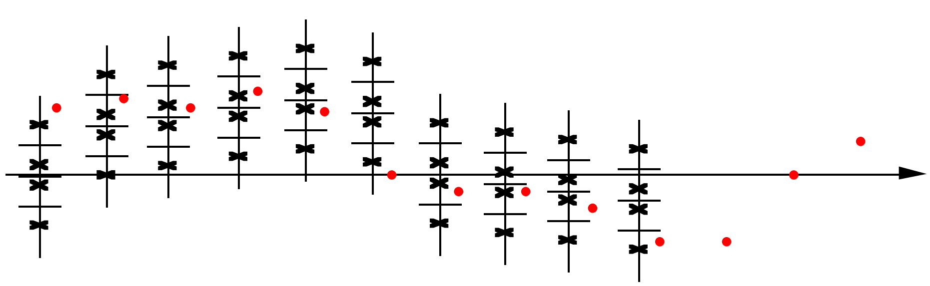


Quantizer Q:



Shifted quantizer b+Q(x-b):





BACKWARD ADAPTIVE QUANTIZATION

- DPCM is a kind of *backward adaptive* quantizer. The quantizer used on X_i is *adapted* based on the reconstructions of past X's.
- It is critical that the rule for adapting how X_i is to be quantized depends only on what the decoder knows. Backward adaptation is a common theme in quantization.
- Other examples of backward adaption

  Quantize X_i with quantizer Q scaled by $|Y_{i-1}|$:

  $$Y_i = |Y_{i-1}|\, Q(X_i/|Y_{i-1}|)\,.$$

  or scaled by the RMS value of recent past Y's

  $$Y_i = A_i\, Q(X_i/A_i)$$

  where

  $$A_i = \left(\frac{1}{M}\sum_{m=1}^{M} Y_{i-m}^2\right)^{1/2}$$

FORWARD ADAPTATION (A DIGRESSION)

Example:

Encoding: Given $X_1,\ldots,X_N$ (e.g. N = 100 or 1000) and scalar quantizers $\{Q_1, Q_2, \ldots, Q_S\}$ each with rate R_o, find s s.t. Q_s gives least distortion on $X_1,\ldots,X_N$, i.e. that minimizes $\Sigma_{i=1}^N (X_i - Q_s(X_i))^2$ (This is not *expected* distortion!)

Encoder output: two components

a. $\lceil \log_2 S \rceil$ bits describing index s of the chosen quantizer.

b. $e_s(X_1),\ldots,e_s(X_N)$, i.e., binary ouput from applying Q_s to each X_i

Rate: $R = R_o + \frac{\lceil \log_2 S \rceil}{N}$ bits/samples

Implementation approaches:

a. Try each quantizer to see which is best, or

b. Use a selection rule, e.g. if quantizers are shifted versions of some basic quantizer, choose the quantizer whose "middle" is closest to $\frac{1}{N}\Sigma_{i=1}^N X_i$.

Design:

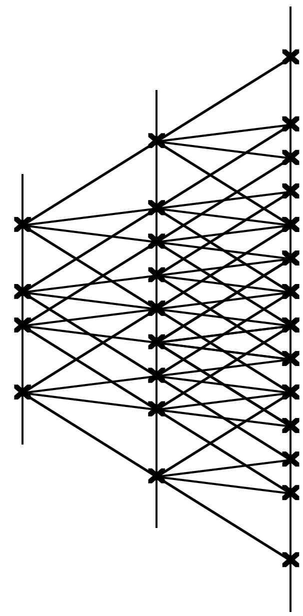
Many possibilities for the Q_s's. Different shifts, scales, point densities, ... One might even design a quantizer from scratch based on $X_1,\ldots,X_N$, and then use the quantizer and lossy encode the quantizer parameters. Choosing N large reduces $\lceil \log_2 K \rceil / N$, but decreases the benefits of forward adaptation.

Which is better, *forward* or *backward* adaption? No conclusive answer.

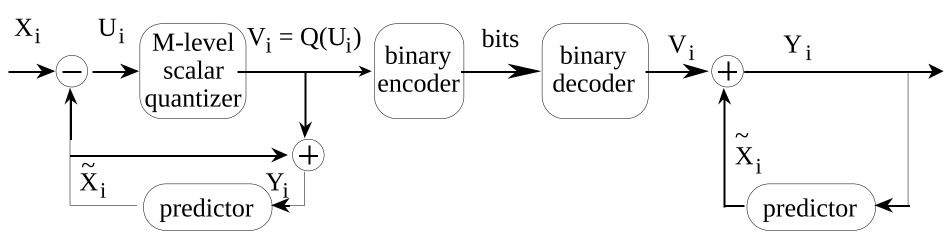
DPCM: THIRD VIEWPOINT -- ENCODER CONTROLS DECODER

- Recall: $Y_i = Y_{i-1} + V_j$ where $V_j \in C = \{w_1,\ldots,w_M\}$ are scalar levels. $Y_0 = 0$.
- Knowing X_i and Y_{i-1}, encoder chooses V_j to be the level w_j that makes $Y_{i-1}+w_j$ closest to X_i; i.e. encoder causes (controls) decoder to make the best possible output.
- This is a greedy encoding algorithm.
- Looking ahead might do better.
- Consider the *tree structure* of DPCM decoder outputs, i.e. of reconstruction sequences shown to the right.
- Tree-encoding: Given N, search the tree for the sequence of N quantization levels in the tree that is closest to the source sequence. Send the indices of the levels in the chosen sequence. Use the usual DPCM decoder.

  This method has never worked its way into practical use. Instead people use the usual DPCM greedy encoder.



DPCM WITH IMPROVED PREDICTION



$\tilde{X}_i = g(Y_{i-1}, Y_{i-2},\ldots)$

- Example of predictor: $\tilde{X}_i = a\,Y_{i-1}$
- Most common: Nth-order linear prediction -- $\tilde{X}_i = \sum_{j=1}^{N} a_j Y_{i-j}$,
  where N = *order* of predictor and $a_1,\ldots,a_N$ are the *prediction coefficients*.
- Assume $Y_0 = 0$
- Key Equations:
  - $Y_i = \tilde{X}_i + Q(X_i-\tilde{X}_i)$ i.e. new reproduction equals old reproduction plus quantized prediction error
  - $X_i-Y_i = U_i - Q(U_i)$ i.e. overall error = error introduced by scalar quantizer $\left(X_i - Y_i = (\tilde{X}_{i-1}+U_i) - (\tilde{X}_{i-1}+Q(U_i)) = U_i - Q(U_i)\right)$
- The system shown above, with finite-order linear prediction, is the usual kind of DPCM. We will presume this in the subsequent discussion.

Performance

Rate:

The rate of DPCM is the rate of the scalar quantizer.

The scalar quantizer could be of the fixed-length or variable-length type. DPCM is fixed rate or variable rate, respectively.

If the scalar quantizer is fixed rate with M levels, then the rate of DPCM is

$$R = \log_2 M$$

If the scalar quantizer is variable-length, then we need to discuss the *nonstationarity* of DPCM before giving a formula for rate.

Distortion:

Since $(X_i - Y_i) = (U_i - Q(U_i))$, the distortion of DPCM is the distortion of the scalar quantizer operating on U. However, before giving a formula for the distortion of DPCM, we need to discuss its nonstationarity.

The nonstationarity of DPCM:

As will be illustrated below, the U_i's are not identical random variables, even when the source to which DPCM is applied is stationary.

Therefore, the mean squared error, $E(X_i-Y_i)^2 = E(U_i-Q(U_i))^2$, changes with i.

Moreover, when variable-length coding is used, the average length of the codewords, $E\, l(U_i)$, changes with i.

Example illustrating nonstationarity:

If the predictor produces $\tilde{X}_i = aY_{i-1}$, then

$U_1 = X_1 - aY_0 = X_1$, $Y_1 = Q(U_1) + aY_0 = Q(X_1)$

$U_2 = X_2 - aY_1 = X_2 - aQ(X_1)$, $Y_2 = Q(U_2)+aY_1 = Q(X_2-aQ(X_1))+aQ(X_1)$

$U_3 = X_3 - aY_2 = X_3 - aQ(Q(X_2-aQ(X_1))+aQ(X_1))$

$U_4 = \cdots$

We can see from the above that the probability distribution of U_i changes with i.

The asymptotic stationarity of DPCM:

It can be shown that, under ordinary conditions, when DPCM is applied to a stationary source X, the U random process is asymptotically stationary. Among other things, this means that the density of U_i converges to some limiting *steady-state* density $f_U(u)$ as $i \to \infty$.

It can also be shown that Y_i and $\tilde{X}_i$ are asymptotically stationary. Moreover, taken jointly, $(X_i, Y_i, \tilde{X}_i, U_i, V_i)$ is asymptotically stationary.

We can now give formulas for the rate of DPCM with variable-length coding and the distortion of DPCM.

Rate with variable-length coding:

When variable-length DPCM is applied to a stationary source, the rate is

$$R = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} E\,L(Q(U_i)) = \lim_{i\to\infty} E\,L(Q(U_i))$$

$$= E\,L(Q(U)) = \int_{-\infty}^{\infty} L(Q(u))\, f_U(u)\, du$$

where $L(v)$ denotes the length of the codeword assigned to quantization level v, and $f_U(u)$ is the asymptotic steady-state density of the U_i's.

Distortion:

When DPCM is applied to a stationary source, the distortion is

$$D = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} E(X_i-Y_i)^2 = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} E(U_i-Q(U_i))^2$$

$$= \lim_{i\to\infty} E(U_i-Q(U_i))^2$$

$$= \int_{-\infty}^{\infty} (u-Q(u))^2\, f_U(u)\, du$$

where is $f_U(u)$ is the steady state density of U.

COMPLEXITY OF DPCM

With fixed-length binary encoding

- Storage: usually quite small
- Arithmetic complexity:

	scalar quantization	prediction	
encoding	R	2N+1	ops/sample
decoding	0	2N+1	ops/sample

- The scalar quantization complexity assumes the scalar quantizer uses binary search to find the closest level to a given input u.

DESIGN OF DPCM

- Design involves the choice of the predictor and the quantizer
- Design of the predictor:

  A first thought is to choose the a_i's to minimize

$$E\,U_i^2 = E\Big(X_i - \sum_{j=1}^{N} a_j Y_{i-j}\Big)^2$$

  We can assume that i is so large that all random variables are characterized by their steady-state probability distributions.

  However, this minimiziation requires knowledge of the covariance matrix of $(Y_{i-1},\ldots,Y_{i-N})$ and correlations $E\,X_iY_{i-1},\ldots,E\,X_iY_{i-N}$, which in turn depend on the a_i's. Thus, we have a chicken-and-egg problem.

  Therefore, the a_i's are usually chosen to minimize

$$E\Big(X_i - \sum_{j=1}^{N} a_j X_{i-j}\Big)^2 = \text{MSPE} = \text{mean square prediction error.}$$

  That is, to predict X_i from $Y_{i-1},\ldots,Y_{i-N}$, use the predictor that would minimize the MSPE if we were predicting X_i from $X_{i-1},\ldots,X_{i-N}$. This is reasonable, because if the DPCM system is working well, then $X_j \cong Y_j$, $j<i$, and so

$$E\,U_i^2 = E\Big(X_i - \sum_{j=1}^{N} a_j Y_{i-j}\Big)^2 \cong E\Big(X_i - \sum_{j=1}^{N} a_j X_{i-j}\Big)^2$$

  This is, essentially, a high-resolution approximation.

- Design of the quantizer:

  A first thought is to design the quantizer to be optimal for the U_i's, i.e. for the limiting density $f_U(u)$. However, we have another chicken-and-egg problem in that the density $f_U(u)$ depends on the quantizer.

  Therefore, the quantizer is usually designed to be optimal for the random variables

$$V_i = X_i - \sum_{j=1}^{N} a_j X_{i-j}$$

  Again we are assuming that i is so large that all random variables are characterized by their steady-state probability distributions.

  As before, this is a reasonable because if the DPCM system is working well, then $X_j \cong Y_j$, $j<i$, and so

$$V_i = X_i - \sum_{j=1}^{N} a_j X_{i-j} \cong X_i - \sum_{j=1}^{N} a_j Y_{i-j} = U_i$$

  Again, this is essentially a high-resolution approximation.

- Designing the predictor and quantizer as stated above is not optimal. But it has stood the test of time, i.e. it seems to be close to optimal.

- One might try to iteratively design a better quantizer (for a given predictor) with the following algorithm

  a. Initialize the quantizer by designing it as described above

  b. Compute or measure the steady-state density $f_U(u)$ of U

  c. Redesign the quantizer to be optimal for $f_U(u)$.

  d. Go to b.

  This seems to produce slightly better quantizers. However, it has not been proven that it converges to the best quantizer for a given predictor, nor even that the distortion of DPCM decreases with each iteration.

- One could use a similar approach to iteratively design a better predictor (for a given quantizer) than the one described previously.

  a. Initialize with the predictor desribed previously.

  b. Measure the relevant steady-state statistics and joint statistics of the X_i's and Y_i's.

  c. Redesign the predictor to be optimal for those statistics.

  d. Go to b.

  This seems to produce slightly better predictors. However, it has not been proven that it converges to the best predictor (for the given quantizer), nor even that the distortion of DPCM decreases with each iteration.

- One can iterate the two design steps given above.

FURTHER NOTES:

- It is not clear that for an optimal DPCM system, Q is the optimal quantizer for the steady-state probability distribution of U. From now on, when we refer to U or its probability distribution, we refer to its steady-state distribution.

  One might think that if Q weren't optimal for U, then one could improve DPCM by making it optimal for U. However, changing Q, changes the pdf of U, so it's not clear that changing Q would help.

- Similarly, it is not clear that for an optimal DPCM system, g is the optimal predictor for X_i based on $Y_{i-N},\ldots,Y_{i-1}$.

  One might think that if g weren't optimal for X, then one could improve DPCM by making it optimal. However, changing g, changes the moments of the Y's, so it's not clear that changing g would help.

- Because the distortion of DPCM is $E(U-Q(U))^2$ and because the steady-state density of U is not easy to compute, there are iterative algorithms for computing actual distortion.

- Additional backward adaptivity can be added to DPCM in various ways, e.g. by adapting the quantizer or the predictor.

- The book by Jayant and Noll is an excellent reference on fancier versions of DPCM.

- DPCM was extensively developed in the 1970's for speech coding. In particular a number of adaptive DPCM methods were developed that adapt the quantizer and/or the predictior. DPCM speech coders are still in use today by telephone companies.
- DPCM is also commonly used today in high performance video coders. In this case, the prediction is from one video frame to the next.
- The Origins of the "PCM" in "DPCM"

  The letters "PCM" comes from "Pulse Code Modulation", which is an old-fashioned term for a modulation technique for transmitting analog sources such as speech. In PCM, an analog source, such as speech, is sampled; the samples are quantized; fixed-length binary codewords are produced; and each bit of each binary codeword determines which of two specific pulses are sent (e.g. one pulse might be the negative of the other, or the pulse might be sinusoidal with different frequencies). Thus PCM is a modulation technique for transmitting analog sources that competes with AM, FM and various forms of pulse modulation. Invented in the 1940's, "PCM" is now viewed as three systems: a sampler, a quantizer and a digital modulator. However, the quantizer by itself is often referred to as PCM. DPCM, i.e. Differential PCM, was originally proposed in the 1940's as an improved PCM type modulation consisting of a sampler, a DPCM quantizer as we know it, and a digital modulator. Nowadays DPCM usually refers just to the quantization part of the system.

- Predictive Coding

  DPCM is also considered to be a kind of *predictive* coding.
- Delta Modulation

  Delta modulation is the special case of DPCM where the quantizer has just two levels: $+\Delta$, $-\Delta$.

  In its most basic form, the predictor produces, simply, $\tilde{X}_i = \hat{X}_{i-1}$. Fancier predictors are also used.

  Delta modulation is used when the source is highly correlated, for example speech coding with a high sampling rate.

High Resolution Analysis of DPCM

- Warning: Though this analysis is almost universally accepted, it has never been satisfactorily proven to be correct.
- Assume source is stationary, zero-mean random process.
- As mentioned earlier, It can be shown that under ordinary conditions $(X_i,Y_i,\tilde{X}_i,U_i,V_i)$ is asymptotically stationary. So we assume it is stationary. Therefore,

$$D = E(X_i-Y_i)^2 = E(U_i-Q(U_i))^2 \qquad \text{(same for all i)}$$

- Key assumption:

  When R is large and quantizer is well designed

$$D \cong E(\tilde{U}_i-Q(\tilde{U}_i))^2$$

  where $\tilde{U}_i = X_i - g(X_{i-1},X_{i-2}, \dots)$.

- Notes:

  $\tilde{U}$ is a stationary, rather than just an asymptotically stationary, random process.

  $\tilde{U}_i \cong U_i$ but $\tilde{U}_i \neq U_i$.

- The key assumption implies

  least possible distortion of DPCM with large rate R

  $\cong$ least possible distortion of SQ with rate R applied to $\tilde{U}$, minimized over choice of predictor g

It follows that for FLC or VLC, and large R

$$\delta_{dpcm}(R) \cong \min_g \delta_{\tilde{U},sq}(R) \cong \frac{1}{12}\left(\min_g \sigma^2_{\tilde{U}}\, \alpha_{\tilde{U}}\right) 2^{-2R}$$

$$\text{where} \quad \alpha_{\tilde{U}} = \begin{cases} \beta_{\tilde{U}}, & \text{for FLC} \\ \eta_{\tilde{U}}, & \text{for VLC} \end{cases}$$

- Conclusions:

  A good (but not necessarily optimal) way to design DPCM is to

  (a) Choose g to minimize $\sigma^2_{\tilde{U}}\, \alpha_{\tilde{U}}$

  (b) Choose Q to achieve $\delta_{\tilde{U},sq}(R)$

- Common situation:

  The pdf of $\tilde{U}$ is similar to a scaled version of that of X, which implies, $\beta_{\tilde{U}} \cong \beta_X$ and $\eta_{\tilde{U}} \cong \eta_X$. Indeed, if X is Gaussian source, then so is $\tilde{U}$, and these approximations become exact. In this case (a) becomes

  (a') Choose g to minimize $\sigma^2_{\tilde{U}}$

  The resulting MSPE is called the N-step prediction error, and is denoted

  $$M_N = \min_g \sigma^2_{\tilde{U}}$$

  We now have the following.

> When R is large and the pdf of $\tilde{U}$ is similar to a scaled version of that of X, e.g. when X is Gaussian, the OPTA function of DPCM with Nth-order prediction is
>
> $$\delta_{dpcm}(N,R) \cong \frac{1}{12} M_N \alpha_X 2^{-2R} \cong \frac{\mathcal{M}_N}{\sigma^2} \delta_{sq}(R) = \frac{1}{G_N} \delta_{sq}(R)$$
>
> $$S_{dpcm}(N,R) \cong S_{sq}(R) + G_N \text{ dB},$$
>
> where
>
> $$G_N = 10 \log_{10} \frac{\sigma^2}{M_N} \text{ dB} = \text{pred'n gain}$$
>
> That is, the gain of DPCM over SQ approximately equals the prediction gain.

MINIMUM MEAN-SQUARED ERROR LINEAR ESTIMATION

- Consider the task of linearly estimating random variable Y from random variables $X_1,\dots,X_N$ with minimum mean squared error (MSE). That is, we wish to choose coefficients $a_1,\dots,a_N$ such that the estimate

  $$\hat{Y} = \sum_{i=1}^{N} a_i X_i$$

  has the smallest possible MSE:

  $$E(Y-\hat{Y})^2 = E\left(Y - \sum_{i=1}^{N} a_i X_i\right)^2$$

- Optimal linear estimators.

  We will now show how to choose the a_i's to minimize the MSE.

  Let us assume that Y and the X_i's have zero mean. If not, we could subtract their means to obtain zero mean random variables, apply optimal linear estimation, and then add the mean of Y to the estimate of Y-EY. Equivalently, we could optimize an estimator of the form $\Sigma_{i=1}^{N} a_i X_i + b$, which is called *affine*.

  We consider $\underline{X} = (X_1,\dots,X_N)^t$ and $\underline{a} = (a_1,\dots,a_N)^t$ to be column vectors. In this vector notation,

  $$\hat{Y} = \underline{a}^t \underline{X}$$

  We assume that the $N \times N$ covariance matrix $K = [EX_iX_j]$ of $\underline{X}$ is known, as is the vector of covariances

  $$\underline{r} = (EX_1Y, \dots, EX_NY)^t$$

ORTHOGONALITY CRITERIA FOR OPTIMALITY

To find the optimal choice of $\underline{a}$ we will prove the following.

- Orthogonality Criteria for Linear Estimators

$\underline{a}$ is the optimal linear estimation coefficients iff

$$E\,(Y-\underline{a}^t\underline{X})X_j = 0, \quad j = 1,\ldots,N$$

i.e. iff the estimation error is orthogonal to all observations. The above equation is called the *orthogonality criteria or principle*.

Proof:

<u>If statement</u>: Suppose $\underline{a}$ satisfies the above, and $\underline{b}$ is an arbitrary set of coefficents. Then,

$$\begin{aligned}
E\,(Y-\underline{b}^t\underline{X})^2 &= E\,\big((Y-\underline{a}^t\underline{X}) + (\underline{a}^t\underline{X}-\underline{b}^t\underline{X})\big)^2 \\
&= E\,(Y-\underline{a}^t\underline{X})^2 + 2\,E\,(Y-\underline{a}^t\underline{X})(\underline{a}^t\underline{X}-\underline{b}^t\underline{X})\big) + E\,(\underline{a}^t\underline{X}-\underline{b}^t\underline{X})^2 \\
&= E\,(Y-\underline{a}^t\underline{X})^2 + 2\,E\,(Y-\underline{a}^t\underline{X})\sum_{j=1}^{N}(a_j-b_j)X_j + E\,(\underline{a}^t\underline{X}-\underline{b}^t\underline{X})^2 \\
&= E\,(Y-\underline{a}^t\underline{X})^2 + 2\sum_{j=1}^{N}(a_j-b_j)\,E\,(Y-\underline{a}^t\underline{X})X_j + E\,(\underline{a}^t\underline{X}-\underline{b}^t\underline{X})^2 \\
&= E\,(Y-\underline{a}^t\underline{X})^2 + E\,(\underline{a}^t\underline{X}-\underline{b}^t\underline{X})^2 \qquad \text{because } E\,(Y-\underline{a}^t\underline{X})X_j = 0 \\
&\geq E\,(Y-\underline{a}^t\underline{X})^2 \qquad \text{because } E\,(\underline{a}^t\underline{X}-\underline{b}^t\underline{X})^2 \geq 0
\end{aligned}$$

Since this is true for every choice of $\underline{b}$, it must be that $\underline{a}$ is optimal. In other words, if $\underline{a}$ satisfies the orthogonality criteria, it is optimal.

<u>Only if statement:</u> Suppose $\underline{b}$ is optimal and $\underline{a}$ satisfies the orthogonality criteria. Then as shown above,

$$E\,(Y-\underline{b}^t\underline{X})^2 = E\,(Y-\underline{a}^t\underline{X})^2 + E\,(\underline{a}^t\underline{X}-\underline{b}^t\underline{X})^2$$

Also, since $\underline{b}$ is optimal

$$E\,(Y-\underline{b}^t\underline{X})^2 \leq E\,(Y-\underline{a}^t\underline{X})^2 .$$

Together the above two equations imply

$$E\,(\underline{a}^t\underline{X}-\underline{b}^t\underline{X})^2 = 0$$

In turn, this implies

$$0 = E\,(\underline{a}^t\underline{X}-\underline{b}^t\underline{X})^2 = E\,(\underline{c}^t\underline{X})^2 = E\,Z^2 \quad \text{where } \underline{c} = \underline{a}-\underline{b} \text{ and } Z = (\underline{a}-\underline{b})^t\,\underline{X}$$

The fact that $E\,Z^2 = 0$ implies $\Pr(Z=0) = 1$, i.e. $Z = (\underline{a}-\underline{b})^t\,\underline{X}$ is always 0. There are two possibilities: One is that $\underline{b} = \underline{a}$, in which case $\underline{b}$ satisfies the orthogonality criteria, since $\underline{a}$ does. The other is that $\underline{b} \neq \underline{a}$, in which case $Z = 0$ means that at least one of the X_i's is a linear combination of the others. In this case, we directly check the orthogonality criteria for $\underline{b}$:

$$\begin{aligned}
E\,(Y-\underline{b}^t\underline{X})X_j &= E\,(Y-(\underline{a}-\underline{c})^t\underline{X})X_j = E\,(Y-\underline{a}^t\underline{X})X_j + E\,\underline{c}^t\underline{X}X_j = \underset{\Uparrow}{0} + E\,ZX_i \\
&= 0 \quad \text{because } \Pr(Z=0) = 1 \qquad (\underline{a} \text{ satisfies orthog crit.})
\end{aligned}$$

which shows that $\underline{b}$ satisfies the orthogonality criteria.

Corollary: Any linear combination of the $\underline{X}_i$'s is orthogonal to the error of an optimal estimator. That is, if $b_1,\ldots,b_K$ are arbitrary numbers, and $\underline{a}$ is the vector of coefficients of an optimal estimator for Y based on $\underline{X}$, then

$$E\,(Y-\underline{a}^t\underline{X}) \sum_{j=1}^{K} b_j X_j = 0$$

<u>Proof:</u> Suppose $b_1,\ldots,b_K$ are arbitrary numbers, and $\underline{a}$ is the vector of coefficients of an optimal estimator for Y based on $\underline{X}$, then

$$E\,(Y-\underline{a}^t\underline{X}) \sum_{j=1}^{K} b_j X_j = \sum_{j=1}^{K} b_j\, E\,(Y-\underline{a}^t\underline{X})\, X_j = 0$$

because orthogonality principle each term $E\,(Y-\underline{a}^t\underline{X})\,X_j$ in sum is zero.

USING THE ORTHOGONALITY PRINCIPLE TO FIND THE OPTIMAL ESTIMATOR

From the orthogonality principle we know that if $\underline{a}$ is the vector of coefficients of the optimal linear estimator, then

$$E(Y-\underline{a}^t\underline{X})X_j = 0, \quad j = 1,\ldots,N$$

This implies

$$E\,YX_j = \underline{a}^t E\,\underline{X}X_j = (E\,\underline{X}X_j)^t\,\underline{a}, \quad j = 1,\ldots,N$$

or equivalently

$$E\,YX_j = (EX_1X_j,\ldots,E\,X_NX_j)\,\underline{a}, \; j = 1,\ldots,N$$

or

$$\begin{bmatrix} EYX_1 \\ \cdot \\ \cdot \\ \cdot \\ EYX_N \end{bmatrix} = \begin{bmatrix} EX_1X_1 \ldots EX_NX_1 \\ \cdots \\ \cdots \\ \cdots \\ EX_1X_N \ldots EX_NX_N \end{bmatrix} \begin{bmatrix} a_1 \\ \cdot \\ \cdot \\ \cdot \\ a_N \end{bmatrix}$$

or

$$\underline{r} = K\underline{a}$$

Where K is the $N \times N$ covariance matrix of $\underline{X}$ and $\underline{r} = E\,Y\underline{X} = (EYX_1,\ldots,EYX_N)^t$.

If K is invertible, i.e. nonsingular, then

$$\underline{a} = K^{-1}\underline{r}$$

is the vector of optimal linear prediction coefficients.

If K is not invertible, i.e. singular, then it can be shown that at least one of the X_i's is a linear combination of the others. Such X_i's can be removed from $\underline{X}$ because any estimator that used them could also be computed from the others. With these X_i's eliminated, the new covariance matrix K' is nonsingular and we can find the optimal $\underline{a}$ by inverting K'.

The MSE of the optimal estimator:

$$MSE = EY^2 - \underline{r}^t K^{-1} \underline{r}$$

Derivation:

$$MSE = E(Y-\underline{a}^t\underline{X})^2 = E(Y-\underline{a}^t\underline{X})(Y-\underline{a}^t\underline{X}) = E(Y-\underline{a}^t\underline{X})Y + E(Y-\underline{a}^t\underline{X})(-\underline{a}^t\underline{X})$$

$= E(Y-\underline{a}^t\underline{X})Y$ by corollary to orthogonality principle

$= EY^2 - \underline{a}^t E\underline{X}Y$

$= EY^2 - \underline{a}^t \underline{r}$

$= EY^2 - \underline{r}^t K^{-1}\underline{r}$ since for optimal estimator $\underline{r} = K\underline{a}$, and so $\underline{a}^t = (K^{-1}\underline{r})^t = \underline{r}^t K^{-1}$ since K & K^{-1} are symmetric

Note: $MSE = EY^2$ is the MSE of a prediction of Y based on no observation.

OPTIMAL LINEAR PREDICTION

Let us now apply what we've learned to the task of designing the optimal predictor of X_{N+1} from $X_1,...,X_N$.

Let us assume that X is a zero-mean, wide-sense stationary random process with autocorrelation function $R_X(k) \equiv EX_iX_{i+k}$.

Linear estimation theory shows that the Nth-order linear predictor of X_i based on $X_{i-1},...,X_{i-N}$ that minimizes the mean squared prediction error has coefficients $\underline{a} = (a_1,\ldots,a_N)^t$ given by

$$\underline{a} = K^{-1}\underline{r}$$

where K is the $N\times N$ correlation/covariance matrix of $X_1,\ldots,X_N$, i.e.

$$K_{ij} = E X_i X_j = R_X(|i-j|), \quad K^{-1} \text{ is its inverse}, \quad \underline{r} = \big(R_X(1),\ldots,R_X(N)\big)^t.$$

It can be shown that if K is not invertible, then the random process X is *deterministic* in the sense that there are coefficients $b_1,...,b_{N-1}$ such that

$$X_n = b_1X_{n-1} + \ldots + b_{N-1}X_{n-N+1} \quad \text{for all } n$$

When K is invertible, the resulting minimum mean square prediction error is

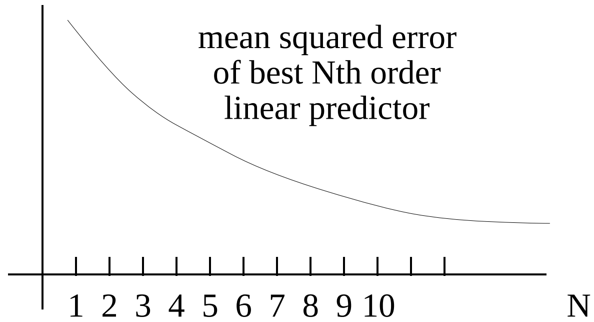
$$M_N = \sigma^2 - \underline{a}^t\underline{r} = \sigma^2 - \underline{r}^tK^{-1}\underline{r} = \text{"N-step prediction error"}$$

and the *prediction gain* is

$$G_N = 10 \log_{10} \frac{\sigma^2}{M_N}$$

BEHAVIOR OF THE N-STEP PREDICTION ERRORS

- Clearly, M_N decreases with N to some limit, as illustrated below, and the prediction gains G_N increase with N to some limit.



- Ot can also be shown that

$$M_N = \frac{|K^{(N+1)}|}{|K^{(N)}|}$$

where $K^{(N)}$ and $K^{(N+1)}$ denote, respectively, the $N \times N$ and $(N+1)\times(N+1)$ covariance matrices of X, and $|K|$ denotes the determinant of the matrix K.

This is derived in supplementary notes.

EXAMPLE: FIRST-ORDER AUTOREGRESSIVE (AR) GAUSSIAN SOURCE.

Assume X is stationary and

$$X_i = \rho X_{i-1} + Z_i \qquad (*)$$

where $-1 < \rho < 1$, Z_i's are IID Gaussian with zero mean and variances σ_Z^2. Z_i is independent of X_j, $j < i$. ($\rho = .9$ is a typical value.) Then

1) $EX_i = 0$:

2) $\sigma_X^2 = EX_i^2 = \frac{\sigma_Z^2}{1-\rho^2}$ $\quad (= 5.26\sigma_Z^2$ if $\rho = .9)$

3) $R_X(k)$ = autocorrelation function $= E\, X_i X_{i+k} = \sigma_X^2 \rho^{|k|}$

$$\rho = \text{correlation coefficent} = \frac{cov(X_i X_{i+1})}{\sigma_X^2} = \frac{R_X(1)}{\sigma_X^2}$$

4) Best predictor for X_i based on $X_{i-1}, X_{i-2}, \ldots, X_{i-N}$ is $\tilde{X}_i = \rho X_{i-1}$ with $M_N = \sigma_Z^2$

Derivation: It's not easy to use $\underline{a} = K^{-1}\underline{r}$. However, we may directly use the orthogonality principle to check that $\tilde{X}_i$ is optimal. Recall that the orthogonal indicates that $\tilde{X}_i$ is optimal if and only if

$$E\,(X_i - \tilde{X}_i)X_{i-j} = 0, \text{ for } j = 1,\ldots,N.$$

We find

$$E\,(X_i - \rho X_{i-1})X_{i-j} = E\,(\rho X_{i-1} + Z_i - \rho X_{i-1})X_{i-j} = E\, Z_i X_{i-j}$$

$$= E\, Z_i\, E\, X_{i-j} = 0, \quad \text{since } Z_i \text{ indep. of past } X\text{'s}$$

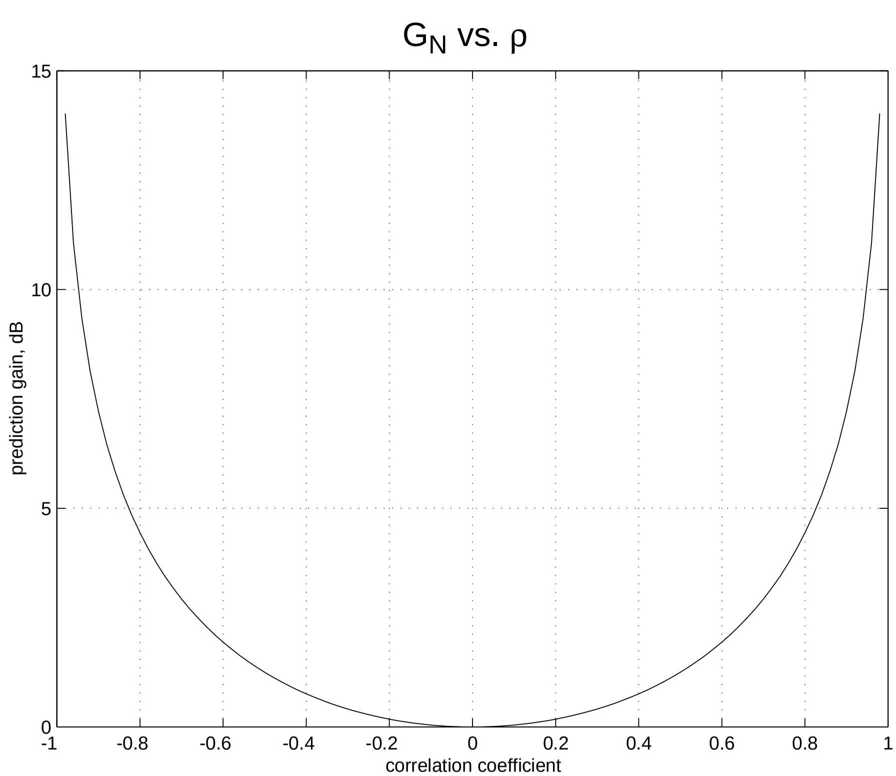
It follows from orthog. principal that $\tilde{X}_i = \rho X_{i-1}$ is optimal.

The MSPE is

$$M_N = E(X_i - \rho X_{i-1})^2 = E(\rho X_{i-1} + Z_i - \rho X_{i-1})^2 = E\, Z_i^2 = \sigma_Z^2$$

Therefore, for $N \geq 1$, the prediction gain (which is the gain of DPCM over SQ) is

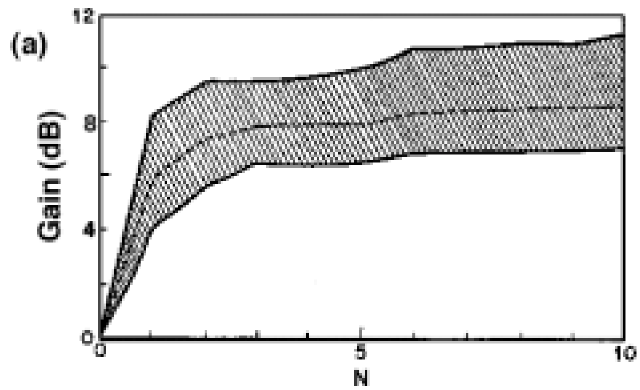
$$G_N = 10 \log_{10} \frac{\sigma_X^2}{M_N} = 10 \log_{10} \frac{1}{1-\rho^2} \qquad (= 7.2 \text{ dB if } \rho = .9)$$



TYPICAL PREDICTION GAINS FOR SPEECH

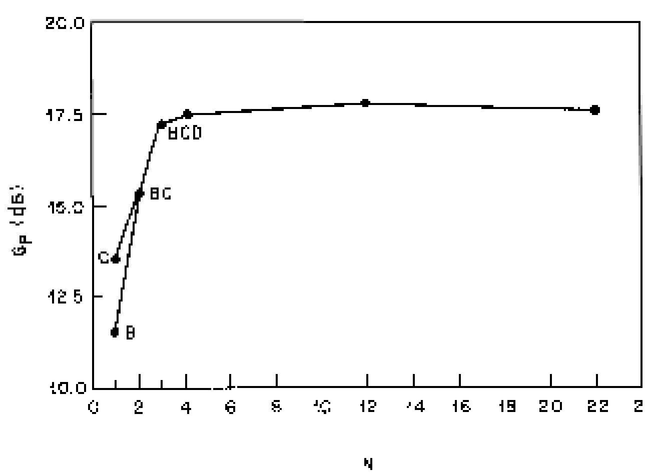
Jayant & Noll, p. 271

For each of several speakers, $R_X(k)$ was empirically estimated and the prediction gains were computed for various N's. The range of G_N values found is shown below. The speech was lowpass filtered, probably to 3kHz.



TYPICAL PREDICTION GAINS FOR IMAGES

Prediction gains in interframe image coding from Jayant and Noll, Fig 6.8, p. 272.



B -- prediction from pixel immediately to the left

$\tilde{X} = .965\, X_B$

C -- prediction from corresponding pixel in previous frame

$\tilde{X} = .977\, X_C$

BC -- prediction from both pixels

$\tilde{X} = .379\, X_B + .617\, X_C$

BCD -- prediction from the two aforementioned pixels plus the pixel to the left of the corresponding in the previous frame

$\tilde{X} = .746\, X_B + .825\, X_C - .594\, X_D$

For larger values of N, the predictor is based on N pixels from the present and the previous frame.

Asymptotic Limit of M_N

For a wide sense stationary random process, it can be shown (see Transform Coding Lectures) that as $N \to \infty$, M_N decreases to

$$\mathcal{Q} = \lim_{N\to\infty} M_N = \text{"one-step prediction error"}$$

$$= \exp\left\{\frac{1}{2\pi}\int_{-\pi}^{\pi} \ln S_X(\omega)\, d\omega\right\}$$

where $S_X(\omega)$ is the power spectral density of X, i.e.

$$S_X(\omega) = \text{discrete-time Fourier transform of } R_X(k)$$

$$= \sum_{k=-\infty}^{\infty} R_X(k)\, e^{-jk\omega}$$

For an Nth-order AR process, $\mathcal{Q} = M_N = M_{N+1} = \ldots$

Asymptotic OPTA of DPCM

In view of the fact that the asymptotic limit of M_N is $\mathcal{Q}$, we have the following.

When R is large and the pdf of $\tilde{U}$ is similar to a scaled version of that of X, e.g. when X is Gaussian, the asymptotic OPTA function of DPCM optimized over N is

$$\delta_{dpcm}(R) \cong \frac{1}{12}\mathcal{Q}\alpha_X\, 2^{-2R} \cong \frac{\mathcal{Q}}{\sigma^2}\,\delta_{sq}(R) = \frac{1}{G_\infty}\,\delta_{sq}(R)$$

$$S_{dpcm}(R) \cong S_{sq}(R) + G_\infty \text{ dB}$$

where

$$G_\infty = 10\log_{10}\frac{\sigma^2}{\mathcal{Q}} \text{ dB} = \text{prediction gain}$$

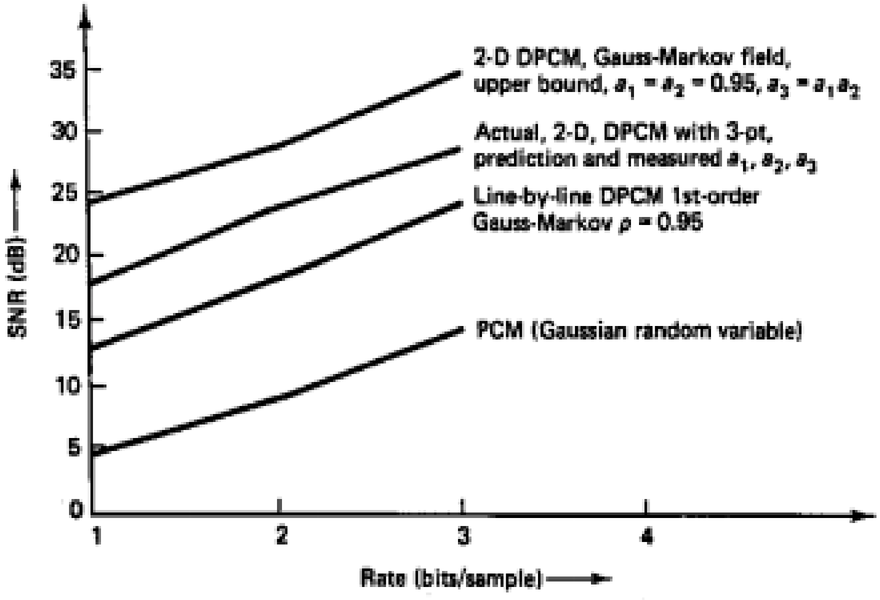
That is, the gain of DPCM over SQ approximately equals the prediction gain.

EXAMPLES OF DPCM IMAGE CODING

Example: Prediction for the present pixel X_{ij}

$$\tilde{X}_{ij} = a_1 Y_{i,j-1} + a_2 Y_{i-1,j} + a_3 Y_{i-1,j-1}$$

$Y_{i-1,j-1}$	$Y_{i-1,j}$
$Y_{i,j-1}$	$X_{i,j}$



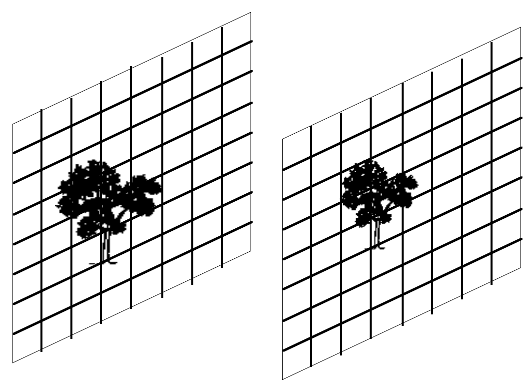
(from R. Jain, Fund'ls of Image Proc, p. 493)

All but second curve from top represents performance predicted with predictor matched to the stated source. For the second curve from the top, the values for a_1, a_2 and a_3 were designed for the actual measured correlations for a test set of images.

DPCM has been extensively studied for image coding. But it is not often used today. Transform coding is more common.

DPCM IN VIDEO CODING

Though not usually called DPCM, the most commonly used video coding methods use a form of DPCM, e.g. MPEG, H.26X, HDTV, satellite "Direct TV", DVD.



Prediction is done on a frame-by-frame basis, with the prediction of a frame being the decoded reconstruction of the previous frame, or a "motion-compensated" version thereof. In the latter case, the coder has a forward-adaptive component in addition to the backward adaption.

Example: (Netravali & Haskell, Digital Pictures, p. 331)

Prediction Coefficients				
left pixel, curr. frame	left pixel prev. frame	same pixel prev frame	MSPE	Pred. gain[1]
1			53.1	12.7 dB
1	-1/2	1/2	29.8	15.3 dB
3/4	-1/2	3/4	27.9	15.5 dB
7/8	-5/8	3/4	26.3	15.8 dB

[1] Based on educated guess that $\sigma^2 = 1000$.

Comparison of DPCM and Transform Coding

Consider a stationary, Gaussian source and large R.

For k-dimensional transform coding:

$$\delta_{tr}(k,R) \cong \frac{1}{12} |K^{(k)}|^{1/k} \alpha_G \, 2^{-2R}$$

For DPCM with kth-order linear prediction

$$\delta_{dpcm}(k,R) \cong \frac{1}{12} M_k \, \alpha_G \, 2^{-2R}$$

where M_k is MSPE of optimal kth-order linear prediction for X_i from $X_{i-k},\ldots,X_{i-1}$.

Fact A: $|K^{(k)}|^{1/k} \geq M_k$.

Proof: $|K^{(k)}|^{1/k} = \left(\sigma_X^2 \prod_{i=1}^{k-1} M_i\right)^{1/k}$ this was proved in the transform coding notes

$\geq M_k$ because all terms being averaged are $\geq M_k$

It follows from this fact that DPCM with kth-order prediction is at least as good as k-dimensional transform coding.

Fact B: $\lim_{k\to\infty} M_k = \lim_{k\to\infty} |K^{(k)}|^{1/k} = \exp\left\{\frac{1}{2\pi}\int_{-\pi}^{\pi} \ln S(\omega)\, d\omega\right\}$

$= \mathcal{Q} =$ 1-step prediction error"

Proof: This was proved in the transform coding notes.

It follows from this fact that for Gaussian sources, large k and large R, DPCM and Transform Coding have the same performance, i.e.

$$\delta_{dpcm}(\infty,R) = \delta_{tr}(\infty,R)$$

However, consider the example of a first-order AR Gaussian source.

Then, since $M_1 = \lim_{k\to\infty} M_k = \mathcal{Q}$,

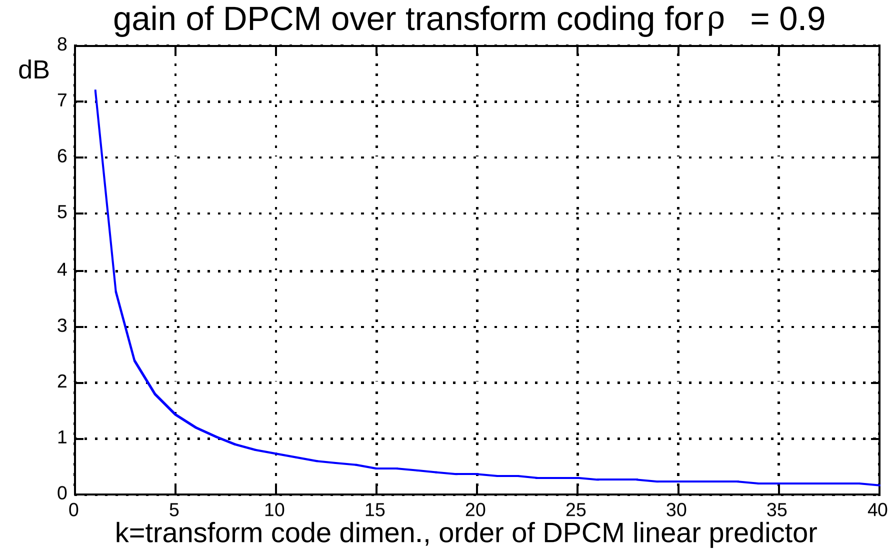
$$\delta_{dpcm}(1,R) \cong \delta_{tr}(\infty,R)$$

Thus, in this case a simple form of DPCM attains performance as good as high dimensional transform coding.

In the Gaussian case, the SNR gain in dB of DPCM with first-order linear prediction over k-dimensional transform coding is

$$10 \log_{10} \frac{\delta_{tr}(k,R)}{\delta_{dpcm}(1,R)} = 10 \log_{10} \frac{(1-\rho^2)^{(k-1)/k}}{1-\rho^2} = -\frac{10}{k} \log_{10} (1-\rho^2)$$

which is plotted below.



Similarly, for an Nth-order AR Gaussian source,

$$\delta_{dpcm}(N,R) \cong \delta_{tr}(\infty,R)$$

- Given that DPCM is more efficient for the AR Gauss source, why is transform coding more commonly used nowadays than DPCM, for example for image coding?

  My feeling is that it because transform coding can more easily be made to take perceptual criteria into account. For example, with transform image coding, it is easy to exploit the fact that the eye is more sensitive to errors in low spatial frequencies than in high spatial frequencies. For example, JPEG does this by using coarser scalar quantizers for high frequency coefficients than for low frequency coefficients.