P-frame: Predicted frame. Coded with reference to the previous frame. 
B-frame: Also a predicted frame. Coded with reference to both previous and future frame.

Coding efficiency: I-frame → P-frame → B-frame

Complexity: I-frame < P-frame < B-frame.

Encoding delay: I-frame < P-frame < B-frame.

b. With layer coder, base layer and enhancement layer are both transmitted. But if base layer is lost or in error, enhancement layer has no use by itself.

2. Since P-frames and predicted frames I-frames can further used, error in P/I-frames could propagate in time.

3. Motion compensation would makes error in reference frame propagate spatially/in time.

2. Variable length coding makes subsequent bits of a errored bit non-decodable.

C. 1. FEC: Forward Error Correction/Detection

2. ARA: Automatic Retransmission Request.

3. Error Resilient encoding: Add redundancy to video bitstreams to assist decoder recovery.

4. Inset more I-frames.

5. Packetizing and slicing.

d. Spatial Scalability: In terms of picture size. User with lower bandwidth can see smaller-size-frames, higher bandwidth for bigger picture.
Temporal Scalability: Frame rate. Higher more frames can be seen.
Amplitude Scalability: (SNR). Higher bandwidth users see the frames which are encoded using smaller quantization stepsize. Users with low bandwidth see frames encoded using larger QP.

e. 1. Size of the object. objects further away looks smaller.
2. Occlusion. If A is occluded by B, A is farther away than B is.
3. Parallax. When moving, objects further away move much slower than objects near us.

f. 1. Basis should be nearly decorrelating.
2. High energy compaction: few of the basis vectors contain big portion of the energy.
3. Easy to compute.
4. Separable: can perform inverse transform and transform.

d. Spectral scalability can be implemented by down
  sampling original frame to smaller frames. Code
  small frames in base layer. Interpolate base
  layer to original size, code interpolated error
  as enhancement layer.
- Temporal scalability can be implemented by
  temporal down sampling. Base layer = base frame
  rate, enhancement layer = skipped frames
- Amplitude scalability by multiple quantizers.
2. a. \[ \mathbf{C} \mathbf{U}_1 = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \]

b. \[ \mathbf{C}_2 = \mathbb{E} \{ \mathbf{A} \mathbf{v} \mathbf{C} \mathbf{v} \mathbf{A}^T \} \begin{bmatrix} \mathbf{A} \\ \mathbf{v} \\ \mathbf{C} \\ \mathbf{v} \end{bmatrix} \]

\[ = \mathbf{e}^2 \begin{bmatrix} \mathbf{P} & \mathbf{P} & \mathbf{P} & \mathbf{P}^2 \\ \mathbf{P} & \mathbf{P} & \mathbf{P} & \mathbf{P} \\ \mathbf{P} & \mathbf{P} & \mathbf{P} & \mathbf{P} \\ \mathbf{P} & \mathbf{P} & \mathbf{P} & \mathbf{P} \end{bmatrix} \]

c. \[ \mathbf{C} \mathbf{L}_2 = \mathbb{E} \{ \mathbf{V} \mathbf{L} \mathbf{C} \mathbf{V}^T \} \quad [\mathbf{V}] := \mathbb{E} \mathbf{L}^T = \mathbf{U} \]

\[ \mathbf{C} \mathbf{L}_2 \mathbf{C} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{P} & \mathbf{P} & \mathbf{P} & \mathbf{P}^2 \\ \mathbf{P} & \mathbf{P} & \mathbf{P} & \mathbf{P} \\ \mathbf{P} & \mathbf{P} & \mathbf{P} & \mathbf{P} \\ \mathbf{P} & \mathbf{P} & \mathbf{P} & \mathbf{P} \end{bmatrix} \mathbf{e}^2 \]

\[ = \mathbf{e}^2 \begin{bmatrix} (\mathbf{P}^2 + \mathbf{P}) & (\mathbf{P}^2 + \mathbf{P}) & (\mathbf{P}^2 + \mathbf{P}) & (\mathbf{P} + \mathbf{P})^2 \\ (\mathbf{P} + \mathbf{P}) & (\mathbf{P}^2 + \mathbf{P}) & (\mathbf{P} + \mathbf{P}) & (\mathbf{P} + \mathbf{P})^2 \\ (\mathbf{P} + \mathbf{P}) & (\mathbf{P} + \mathbf{P}) & (\mathbf{P}^2 + \mathbf{P}) & (\mathbf{P} + \mathbf{P})^2 \\ (\mathbf{P} + \mathbf{P}) & (\mathbf{P} + \mathbf{P}) & (\mathbf{P} + \mathbf{P}) & (\mathbf{P}^2 + \mathbf{P})^2 \end{bmatrix} \]

\[ \mathbf{C} \mathbf{L}_2 \mathbf{C}^T = \frac{1}{2} \begin{bmatrix} (\mathbf{P}^2 + \mathbf{P}) & (\mathbf{P}^2 + \mathbf{P}) & (\mathbf{P}^2 + \mathbf{P}) & (\mathbf{P}^2 + \mathbf{P}) \\ (\mathbf{P} + \mathbf{P}) & (\mathbf{P}^2 + \mathbf{P}) & (\mathbf{P} + \mathbf{P}) & (\mathbf{P} + \mathbf{P}) \\ (\mathbf{P} + \mathbf{P}) & (\mathbf{P} + \mathbf{P}) & (\mathbf{P}^2 + \mathbf{P}) & (\mathbf{P}^2 + \mathbf{P}) \\ (\mathbf{P} + \mathbf{P}) & (\mathbf{P} + \mathbf{P}) & (\mathbf{P} + \mathbf{P}) & (\mathbf{P}^2 + \mathbf{P}) \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \mathbf{e}^2 \]

\[ = \mathbf{e}^2 \begin{bmatrix} 4(\mathbf{P}^2 + \mathbf{P}) & 0 & 0 & 0 \\ 0 & 4(\mathbf{P} + \mathbf{P}) & 0 & 0 \\ 0 & 0 & 4(\mathbf{P} + \mathbf{P}) & 0 \\ 0 & 0 & 0 & 4(\mathbf{P} + \mathbf{P}) \end{bmatrix} \]
d. 
\[ E_{t,k} = \frac{1}{2} (p^t p^k, (1-p^t), (1-p^k), (p^{-t} p^{-k})) \]

\[ \left( \prod_k E_{t,k} E_{t,k}^2 \right)^{1/4} = \left( E_{t,1} E_{t,2} E_{t,3} E_{t,4} \right)^{1/4} \cdot (p^t)^{1/2} (1-p^t) (1-p^k) (p^{-t}) \cdot (1-p^{-k}) \]

\[ \frac{R_1}{2} = R + \frac{1}{2} \log_2 \left( \frac{E_{t,1} (p^t)}{\prod_k E_{t,k}^4} \right) = R + \frac{1}{2} \log_2 \left( \frac{E_{t,1} (p^t)}{(1-p^t)^4} \right) \]

\[ R_2 = R + \frac{1}{2} \log_2 \left( \frac{E_{t,2}^2}{\prod_k E_{t,k}^4} \right) \]

\[ R_3 = R + \frac{1}{2} \log_2 \left( \frac{E_{t,3}^2}{\prod_k E_{t,k}^4} \right) \cdot (1-p^t) \]

\[ R_4 = R + \frac{1}{2} \log_2 \left( \frac{E_{t,4}^2}{\prod_k E_{t,k}^4} \right) (p^t) \]

Average distortion: 
\[ D_0 = \frac{1}{2} \left( E_{t,1} (p^t) \right)^2 \cdot 2^{-2R_1} + E_{t,2} (1-p^t) \cdot 2^{-2R_2} + \frac{1}{2} E_{t,3} (1-p^t) \cdot 2^{-2R_3} + E_{t,4} (p^t) \cdot 2^{-2R_4} \]

\[ = 2^{-2R} \cdot (\prod_k E_{t,k}^4) (1-p^t) \cdot 2^{-2R} \]

e. 
\[ D_5 = E_{t,5} \cdot 2^{-2R} \]

f. 
\[ C_{tk} = \frac{D_5}{D_t} = \frac{E_{t,5}^2}{\left( \prod_k E_{t,k}^4 \right) (1-p^t) \cdot 2^{-2R}} = \frac{E_{t,5}^2}{\left( \prod_k E_{t,k}^4 \right)} \cdot \frac{1}{(1-p^t) \cdot 2^{-2R}} = \frac{1}{1-p^t} \]

Could have assumed: 
\[ E_{t} = E_{t+1} = \frac{3}{1-p^t} \]
3. a. \( E f_n(x,y) f_{n-1}(x,y)^2 = \rho \delta^2 \)

\[ E f_n(x,y) f_{n-1}(x,y)^2 = E f_n(x,y) f_{n-1}(x,y)^2 \]

\[ \mathcal{R}(1,1)^2 = [\mathcal{R}(0,1)]^2 \]

\[ \mathcal{R}(1,1)^2 [\mathbf{a}] = [\mathcal{R}(0,1)]^2 [\mathbf{a}] \]

\[ \mathcal{R}(1,1)^2 [\mathbf{b}] = [\mathcal{R}(0,1)]^2 [\mathbf{b}] \]

\[ b_1 = b_2 = \frac{\rho}{1 + \rho^2} \]

\[ \delta_\rho = \delta_\rho \left( 1 - \left[ \rho \rho \left[ \frac{H_\rho}{H_\rho} \right] \right] \right) \]

\[ = \delta_\rho \frac{1 - \rho^2}{1 + \rho^2} \]

C. For the \( R \)-mode:

\[ D_\rho = \delta_\rho \left( 1 - \rho^2 \right) \delta_\rho \cdot 2^{-\alpha} R_\rho \]

\[ R_\rho = \frac{1}{\alpha} \log \frac{D_\rho}{\delta_\rho (1 + \rho^2)^2} = \frac{1}{\alpha} \log \frac{\frac{3}{2} (1 - \rho^2)^{1/2}}{D_\rho} \]

For the \( B \)-mode:

\[ D_\rho = \delta_\rho \frac{H_\rho}{H_\rho} \delta_\rho \cdot 2^{-\alpha} R_\rho \]

\[ R_\rho = \frac{1}{\alpha} \log \frac{D_\rho (1 + \rho^2)^2}{\delta_\rho (1 - \rho^2)^{1/2}} \leq R_\rho \]

\[ = \frac{1}{\alpha} \log \frac{\frac{3}{2} (1 - \rho^2)^{1/2}}{D_\rho} \]

d. Because \( B \)-mode requires extra delay, since it needs future frame to encode current frame. In most real applications, e.g. live streaming, video conferencing, video is very sensitive to delay. Thus \( B \)-mode is not used.
4. a. \( x_i = \frac{X + B/2}{Z} \)  
   \( X_r = \frac{X - B/2}{Z} \)

\[ x_i = \frac{X + B/2}{Z} \]

\[ X_r = \frac{X - B/2}{Z} \]

\[ Y = \frac{y + B}{Z} \]

\[ x_i - X_r = \frac{EB}{Z} \]

\[ Z = \frac{F \cdot B}{x_i - X_r} \]

\[ x_i + X_r = 2F \cdot \frac{x_i}{Z} \]

\[ x = \frac{(x_i + X_r) \cdot Z}{2F} = \frac{(x_i + X_r) \cdot F \cdot B}{2(x_i - X_r) \cdot F} = \frac{x_i + X_r}{2(x_i - X_r)} \cdot B \]

\[ Y = y \cdot \frac{Z}{F} = y \cdot \frac{B}{x_i - X_r} \]

b. \( x_i = F \cdot \frac{x + B/4}{Z} \)

\[ x_i = F \cdot \frac{x_i + X_r + \frac{B}{4}}{2(x_i - X_r)} \]

\[ Y_i = \frac{F \cdot Y}{Z} = y \]

C. 1. Divide the image into blocks, for each block, assume it is a flat surface, so every point in it has some disparity.

2. For each block in the right image, minimize the disparity error:

\[ E = \sum_{i,j} (f_r(x_i,y_j) - f_l(x_i + d(x_i,y_j), y_j))^2 \]

Get the disparity \( d(x_i,y_j) \).

3. For every block in the intermediate image:

   For each pixel \((x_i,y_j)\), compute corresponding \( X_r \) using:

   \[ x_i = \frac{X + B/4}{Z} \]

   \[ X_r = \frac{X - B/4}{Z} \]

   Then set \( f(x_i,y_j) = f(x_i,y_j) \).
5. function [QP] = Encode (F, PF, width, height, minmax, maxmax, AS, outfile)

    QP = zeros(width/height). For C = 1:8:height) for (c = 1:8:width) % for each 8x8 block
    Block = F(c; t+7, c; c+7);
    % first do backward motion estimation
    [muh, mwv, Predicted Block] = MotionEstimation (Block, PF, width, height, minmax, maxmax);
    SAD_P = sum (sum (abs (Block - Predicted Block)));
    % code directly.
    SAD_I = sum (sum (abs (Block - 128)));
    if SAD_P > SAD_I mode = 1; % code directly.
    else mode = 0; end;

    if mode == 0 % compute
        % DCT coefficients of the error block.
        if mode == 0 % predicted
            ErrBlock = Block - Predicted Block;
            muh = 0, mwv = 0;
        else ErrBlock = Block - 128; end;

        DCT Block = dct2 (ErrBlock);
        [Quantized DCT Block] = quantizeDCT (DCT Block, AS); % defined below
        EntropyCoding (mode, muh, mwv, Quantized DCT Block, outfile);
    end;
    end;
    end;
    end;
    end;

    function [Quantized DCT Block] = quantizeDCT (DCT Block, AS)
    % Quantization of DCT coefficients is defined:
    function [Quantized DCT Block] = quantizeDCT (DCT Block, AS)
Quantized DCT Block $= \text{floor}(\text{DCT Block} / Q_s)$

Need to shift DCT Block $+ Q_s / 2$

dequantize DCT function:

function [deQuantizedDCTBlock] = dequantizeDCT(DCT Block, Qs)

decompressedDCTBlock = DCT Block * Qs;