Q1. (35pts)

a) What is the relation between the spectrum of a light source and perceived color?

b) What are the conditions for the existence of normal flow? Can we always recover optical flow from normal flow? Discuss the relation between the spatial image gradients and the aperture problem.

c) Discuss the relationship between the MAP motion estimator and Horn and Shunk algorithm.

d) Find the spatial temporal impulse response of the filter that performs “merging” of even and odd fields to form a progressive frame. Find the frequency response of the filter. Discuss the frequency domain interpretation of merging for stationary and moving image regions.

e) Suppose you have a grayscale image of NxM. Compare the computational requirements of an exhaustive block matching (integer pel) algorithm with block size of KxK and two-level hierarchical block matching algorithm. The maximum motion range is +/-L. For comparison use the number of operations where each operation includes 1 subtraction, 1 addition and 1 absolute value operation.

Q2 (35 pts) Consider the conversion of a Bt.601 (720x480 – interlaced) 4:2:2 video signal to 4:2:2. Y component is the same in these formats and Cr, Cb components are represented as

```
<table>
<thead>
<tr>
<th></th>
<th>f1</th>
<th>f2</th>
</tr>
</thead>
<tbody>
<tr>
<td>4:2:2 format</td>
<td>f1</td>
<td>f2</td>
</tr>
<tr>
<td>4:2:0 format</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

(f stands for field, x for Cr and o for Cb)
Illustrate the sampling lattices in both formats and determine the intermediate lattice and ideal interpolation filter.

Q3 (30 pts)

Assume a source consists of 4 symbols. The probabilities of the symbols:

\[ p(a_1) = 0.5, \ p(a_2) = 0.2, \ p(a_3) = 0.2, \ p(a_4) = 0.1 \]

a) Compute the entropy of the source. Apply Huffman coding and compute the average bitrate.

b) Assume, we further know that the conditional probability distribution \( q(i|j) \) as:

\[
\begin{bmatrix}
0.6 & 0.4 & 0.4 & 0.4 \\
0.2 & 0.3 & 0.2 & 0.2 \\
0.1 & 0.2 & 0.3 & 0.1 \\
0.1 & 0.1 & 0.1 & 0.3 \\
\end{bmatrix}
\]

The joint probability function of every two symbols can be determined by

\[ p(a_i, a_j) = p(a_i)q(i|j) \]

Applying this to all possible combinations of two symbols, one can obtain the probabilities of all possible 2-D vector symbols. Apply Huffman coding to this new source. Compute the average bitrate. Compare your result with part a.
Q1 (50 pts) Motion Estimation (ME) is an important part of any video compression system. Unfortunately it is also the most computationally intensive function of the entire encoding process. In motion estimation the current image is divided into Macro-Blocks (MB) and for each MB, a similar one is chosen in a reference frame, minimizing a distortion measure. The best match found represents the predicted MB, while the displacement from the original MB to the best match gives the so-called Motion Vector (MV). Only the MV and the residual (i.e. the difference between the original MB and the predicted MB) need to be encoded and transmitted into the final stream. The distortion measure can be chosen as the Sum of Absolute Differences. Full Search Block-Matching (FSBM) motion estimation is the technique suggested in the reference software models of most of the video coding standards however it is computationally very intensive.

a) Find out the computational complexity of FSBM for a given image size, block size and search window.

b) Come up with a new algorithm to reduce this computational complexity while still using block matching. You can modify the block size, search criteria etc.

c) Implement your proposed method and show visual quality in terms of PSNR of the motion compensated image. You can use two consecutive frames of any video sequence to show the results.

Q2 (40 pts) Consider the conversion of a BT.601 (720x480 – interlaced) 4:2:2 video signal to 4:2:0. Illustrate the sampling lattices in both formats and determine the intermediate lattice and ideal interpolation filter.

Q3 (10 pts) Encode and decode the following sequence using Huffman and arithmetic coding. Calculate the bit rates for each case. You can use the occurrence frequency of each symbol in the sequence as the estimate of the probability of the symbol.

Sequence: a d d a a c c c b a b a b a c