6.1 Describe the pros and cons of different motion representation methods (pixel-based, block-based, mesh-based, global).

- Pixel-based
  - Pros: Universally applicable
  - Cons: Requires estimation of many unknowns, requires physical constraints
- Block-based
  - Pros: Fixed partitions require no manual determination of regions, fewer unknowns than a pixel-based approach
  - Cons: Motion is discontinuous across block boundaries, causing blocking artifacts
- Mesh-based
  - Pros: Continuous motion across an object's surface
  - Cons: Fails to capture motion discontinuities at boundaries between objects
- Global
  - Pros: Describes motion using few parameters
  - Cons: Is not adequate if there are multiple directions of motion in a scene

6.10 Consider a CCIR601 format video, with Y-component frame size of 720×480. Compare the required computation by an EBMA algorithm (integer-pel) with block size 16×16 and that by a two-level HBMA algorithm. Assume the maximum motion range is ± 32. You can compare the computation by the operation number with each operation including one subtraction, one absolute value computation, and one addition. You can make your own assumption about the search range at different levels with HBMA. For simplicity, ignore the computations required for generating the pyramid and assume only integer-pel search.

The required computation for EBMA can be broken down into several subsets of calculations:

1. The required computation to compare an anchor block to a target block using the MAD criterion.
2. The number of block comparisons with an entire search window. (One search window per anchor block.)
3. The number of anchor blocks within an entire anchor frame.

These subsets are summarized in the formulas below:

Computations for EBMA:

$$\text{Block size } N \times N \quad \Rightarrow N^2 \text{ calculations for MAD}$$

$$\text{Window} \quad \pm R \quad \Rightarrow \text{candidate blocks } (2R+1)^2$$

$$\text{Image} \quad M \times M \quad \Rightarrow \left(\frac{M}{N}\right)^2 \text{ blocks}$$

$$\textit{Computations Total} \Rightarrow \frac{M^2}{N^2} N^2 (2R+1)^2 = M^2 (2R+1)^2$$

The specifications given in the question suggest N = 16 and R = 32. The image size, however, is not square, so our values are M_1 = 720 and M_2 = 480. This leads to a computation total of 720 x 480 x $(2*32+1)^2$ = 1,460,160,000 computations.

6.12 Write a C or *Matlab* code for implementing EBMA with integer-pel accuracy. Use a block size of 16 × 16. The program should allow the user to choose the search range, so that you can compare the results obtained with different search ranges. Note that the proper search range depends on the extent of the motion in your test images. Apply the program to two adjacent frames of a video sequence. Your program should produce and plot the estimated motion field, the predicted frame, the prediction error image. It should also calculate the PSNR of the predicted frame compared to the original tracked frame. With *Matlab*, you can plot the motion field using the function *quiver*.

Results of this program using a 5x5 search window are displayed below.

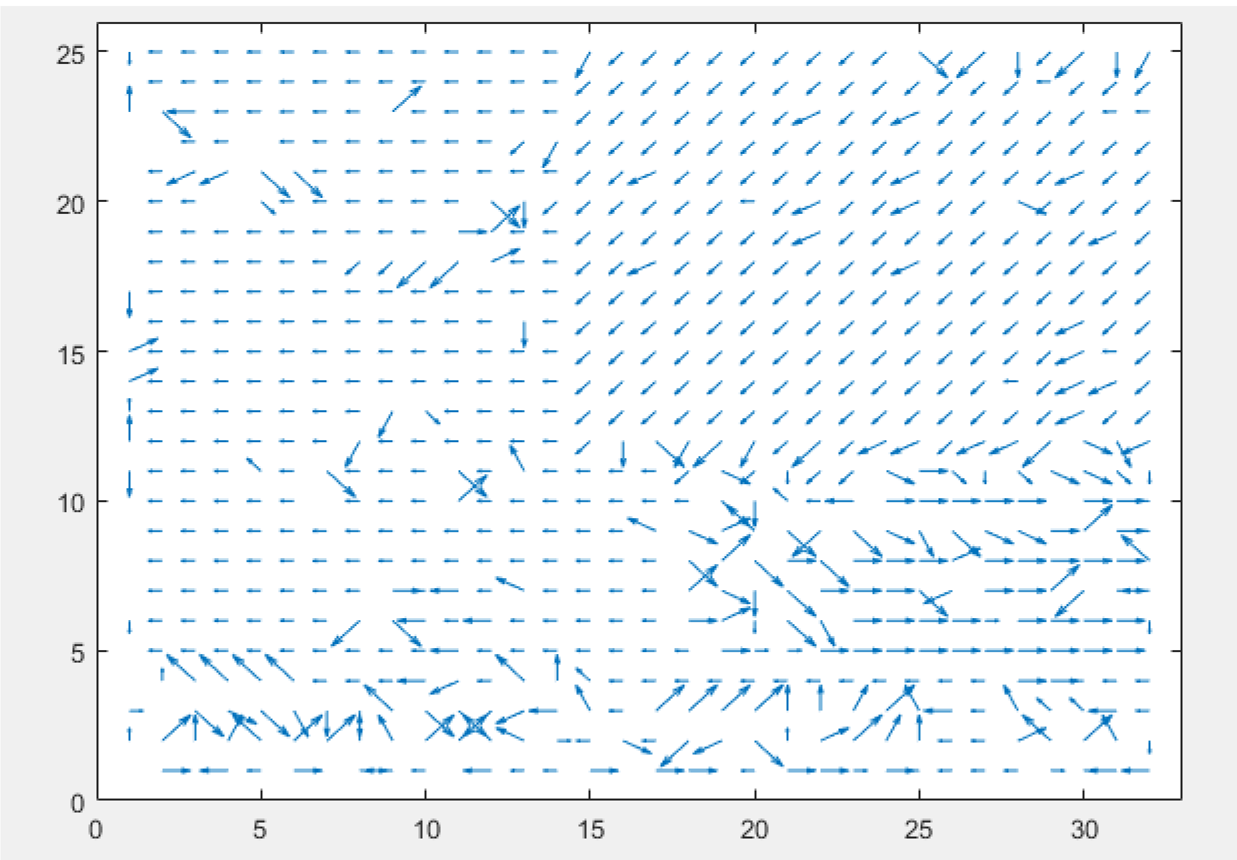

Figure 1: Estimated Motion Field

Figure 2: Predicted Frame

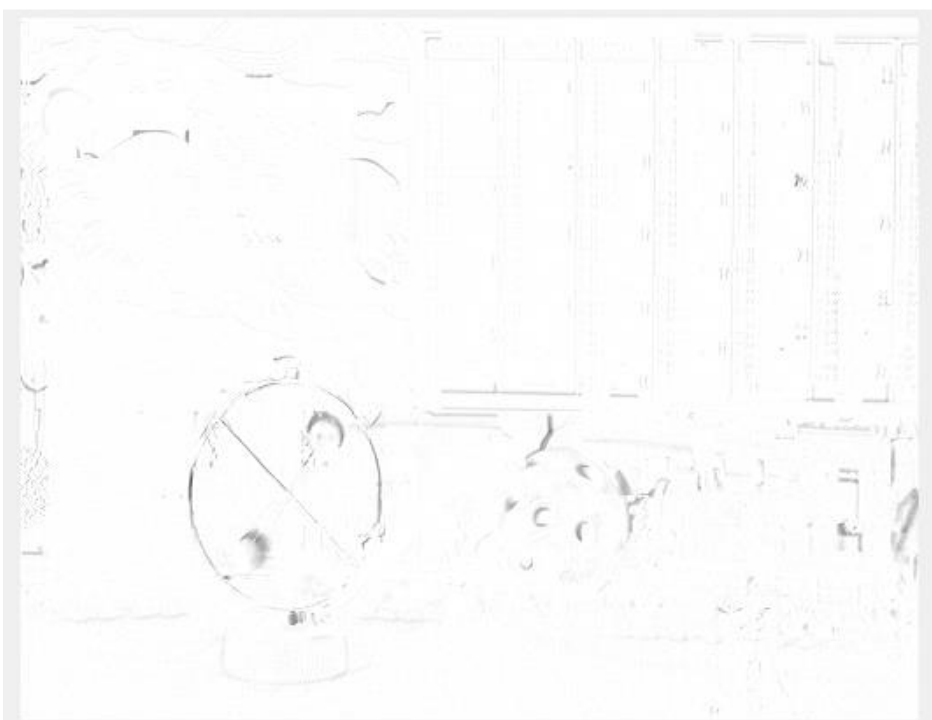

Figure 3: Error frame, complemented for easy viewing