Lecture 7, Video Coding,
Motion Compensation Accuracy

Last time we saw several methods to obtain a good motion estimation, with reduced complexity (efficient search), and with the possibility of sub-pixel accuracy.
Question: How does the search area, the search process and the accuracy affect the **compression performance?**
A comparison of different search window sizes for the resulting total **sum of absulute errors** (SAE) can be found in Book Richardson: Video Codec Design. There a comparison with an example video was made, with 16x16 pixel block sizes for the motion compensation (full search motion estimation).

-**without motion compensation**, just difference between 2 consecutive frames: Total SAE= **1326783**. This is an estimate of the upper limit of the prediction error for motion compensation (a starting point here).
-**With motion compensation**, **search window** +/-1 pixel (minimum complexity) : SAE **1 278 610,** a small improvement, but the number of operations needed for the motion compesation is: **1x10^6,** hence fairly high
-Serarch window +/- 3 pixel: SAE **1 173 060**, somewhat lower, but we considerbly increased the complexity: **5.2x10^6**

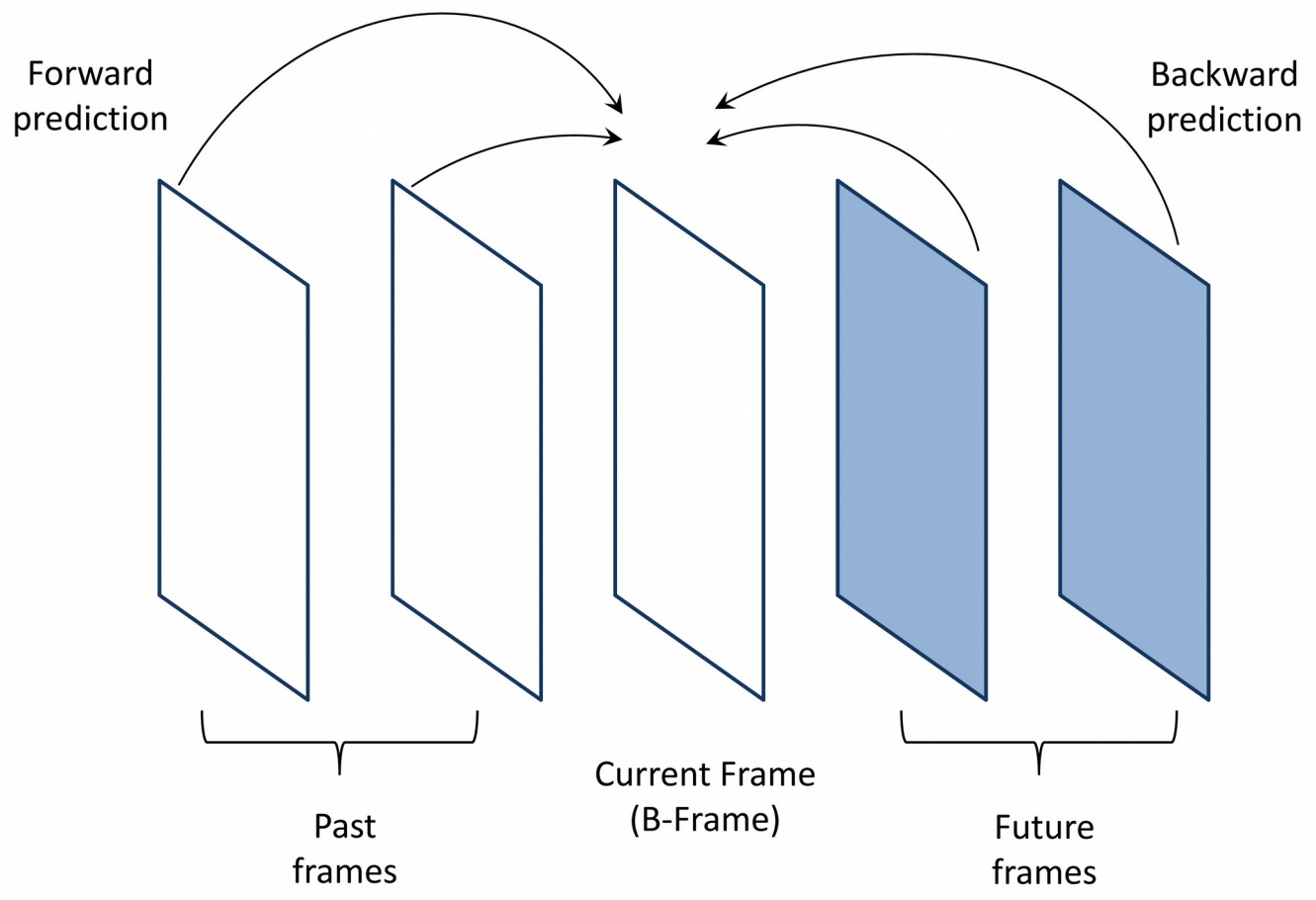
-Search window +/- 7: SAE **898 581**, complexity: **23.4x10^6**
-Search window +/- 15 pixel: SAE **897 163**, not much improvement anymore, but a lot of increase in complexity: **99.1x10^6**
This was for the most complex search method (full search). How does this compare with a **more efficient search method?** Again we can find a comparison for the same video sequence in this reference, with a search window of +/- 15:

Full search: SAE **897 163**, complexity: **99.1x 10^6**
Three-step search: SAE **914753**, complexity: **3.6x10^6**!

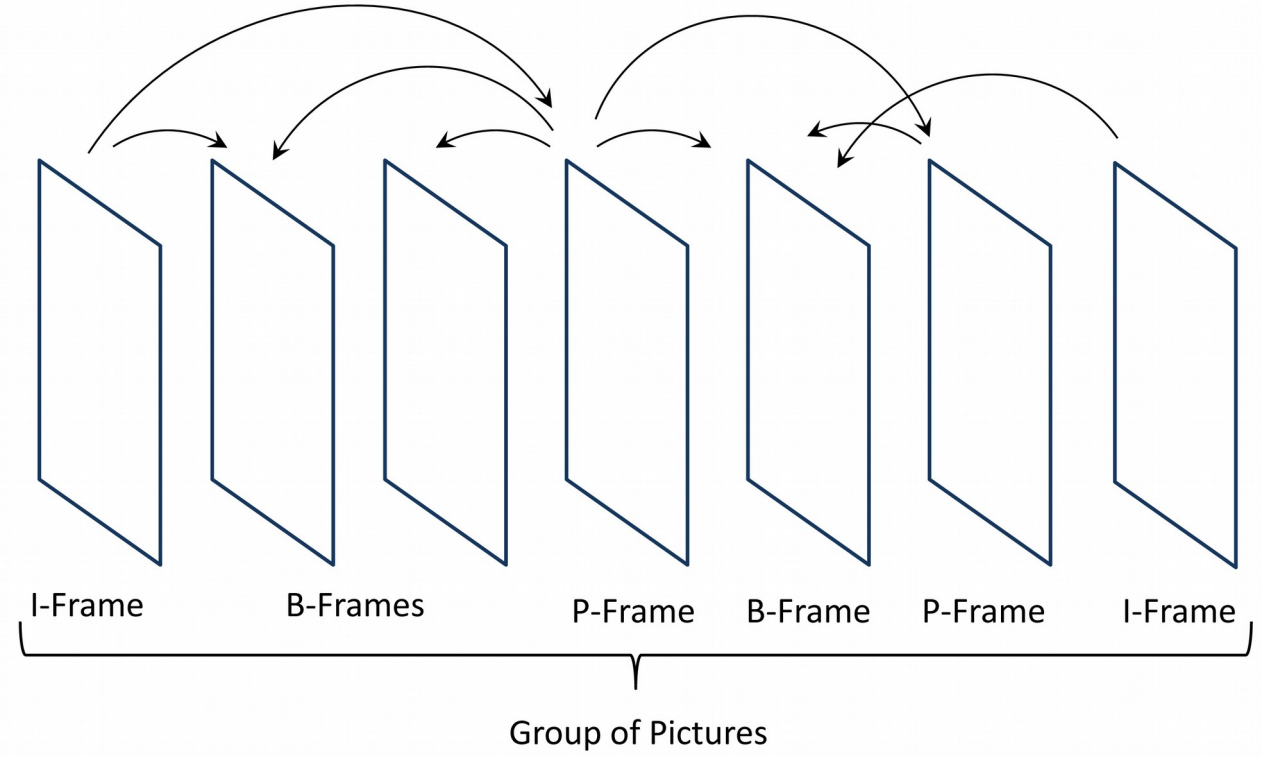
Here we can see that we **gain a lot of complexity reduction** using the three-step search (the hierarchical approach), but only **pay a small price** in increased SAE!

Choice of Reference Frames
-**Forward prediction (P-Frames)**: prediction is based on past frames (also more than 1 frame in the past). Advantage: we don't need any additional delay for our prediction. Disadvantage: For instance for scene cuts, it doesn't work for the following frame, because of missing similarity.

-**Backwards prediction**: prediction based on future frames.
Advantage: can be used if there is no or not sufficient similarity to the past frame(s).
Disadvantage: We need a **delay** to wait for the future frame to predict from.
-**Bidirectional prediction (B-Frames)**: We use both directions to predict the current frame (weighted average of both directions, or taking the best of both). For each macroblock of the motion compensation the encoder transmits a mode information:
Mode a: forward prediction
Mode b: backward prediction
Mode c: average of the 2 predictions.
Advantage: We obtain the best prediction because we have the most information as a basis for it. It improves the compression efficiency by up to 50%.
Disadvantage: It needs side information for the signaling, and it needs delay for the future frame.
- **I-Frames (Intra Frames):** No prediction using motion compensation :
Advantage: independent of other frames, can be used to start decoding the video sequence.
Disadvantage: Need the highest bit-rate.



In a typical video coding sequence (for instance MPEG video), those different types of frames are grouped in a so-called Group of Pictures (GOP). The group of pictures is bounded by I-frames on both sides, and contains B-frames and P-frames in between. For a good compression performance, we should choose only a few I-Frames, many B-frames, and a few P-Frames. A video sequence conists of many consecutive GOPs.



Observe that predicted frames can only reference already reconstructed frames. That is why referencing B-Frames is problematic, because they are reconstructed last.

An example is
I,P,B,B,I,P,B,B,I....
which shows that each GoP is separated by an I-Frame from the other.

Enhancements to the Motion Model

Variable Block sizes: depending on the image content, it might useful to use different block sizes, for instance 16x16 pixels, 8x8 , or 4x4, and ideally this would be adapted to the

image content. This is used for instance in H.263, H.264 and H.265. The latters also offers the flexibility of sub-dividing those blocks into rectangular sub-blocks (4x8, 8x16,...).

Complex Motion Models: Picture warping can be used to take into account the effect of zooming in our out, or rotations of objects. But it also leads to increased complexity in encoder and also decoder.

Object based coding and motion compensation: here we don't have blocks, but find out the boundaries of our moving objects in the scene, and apply motion compensation to it. Theoretically this is very efficient for the motion compensation, but very computationally complex especially for the encoder. This is one of the reasons this approach is still an active research area.