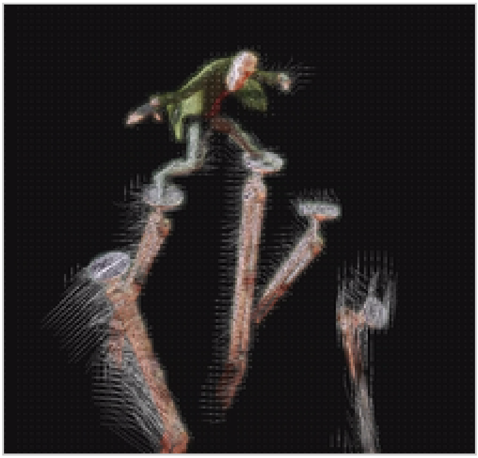
Motion compensation

From Wikipedia, the free encyclopedia

Motion compensation is an algorithmic technique used to predict a frame in a video, given the previous and/or future frames by accounting for motion of the camera and/or objects in the video. It is employed in the encoding of video data for video compression, for example in the generation of MPEG-2 files. Motion compensation describes a picture in terms of the transformation of a reference picture to the current picture. The reference picture may be previous in time or even from the future. When images can be accurately synthesised from previously transmitted/stored images, the compression efficiency can be improved.



Visualization of MPEG block motion compensation. Blocks that moved from one frame to the next are shown as white arrows, making the motions of the different platforms and the character clearly visible.

Contents

How it works

Motion compensation exploits the fact that, often, for many frames of a movie, the only difference between one frame and another is the result of either the camera moving or an object in the frame moving. In reference to a video file, this means much of the information that represents one frame will be the same as the information used in the next frame.

Using motion compensation, a video stream will contain some full (reference) frames; then the only information stored for the frames in between would be the information needed to transform the previous frame into the next frame.

Illustrated example

The following is a simplistic illustrated explanation of how motion compensation works. Two successive frames were captured from the movie Elephants Dream. As can be seen from the images, the bottom (motion compensated) difference between two frames contains significantly less detail than the prior images, and thus compresses much better than the rest. Thus the information that is required to encode compensated frame will be much smaller than with the difference frame. This also means that it is also possible to encode the information using difference image at a cost of less compression efficiency but by saving coding complexity without motion compensated coding; as a matter of fact that motion compensated coding (together with motion estimation, motion compensation) occupies more than 90% of encoding complexity.

Type	Example Frame	Description
Original		Full original frame, as shown on screen.
Difference		Differences between the original frame and the next frame.
Motion compensated difference		Differences between the original frame and the next frame, shifted right by 2 pixels. Shifting the frame *compensates* for the panning of the camera, thus there is greater overlap between the two frames.

Motion Compensation in MPEG

In MPEG, images are predicted from previous frames (P frames) or bidirectionally from previous and future frames (B frames). B frames are more complex because the image sequence must be transmitted/stored out of order so that the future frame is available to generate the B frames.[1]

After predicting frames using motion compensation, the coder finds the error (residual) which is then compressed and transmitted.

Global motion compensation

In **global motion compensation**, the motion model basically reflects camera motions such as:

- Dolly - moving the camera forward or backward
- Track - moving the camera left or right
- Boom - moving the camera up or down
- Pan - rotating the camera around its Y axis, moving the view left or right
- Tilt - rotating the camera around its X axis, moving the view up or down
- Roll - rotating the camera around the view axis

It works best for still scenes without moving objects.

There are several advantages of global motion compensation:

- It models the dominant motion usually found in video sequences with just a few parameters. The share in bit-rate of these parameters is negligible.
- It does not partition the frames. This avoids artifacts at partition borders.
- A straight line (in the time direction) of pixels with equal spatial positions in the frame corresponds to a continuously moving point in the real scene. Other MC schemes introduce discontinuities in the time direction.

MPEG-4 ASP supports GMC with three reference points, although some implementations can only make use of one. A single reference point only allows for translational motion which for its relatively large performance cost provides little advantage over block based motion compensation.

Moving objects within a frame are not sufficiently represented by global motion compensation. Thus, local motion estimation is also needed.

Block motion compensation

In **block motion compensation** (BMC), the frames are partitioned in blocks of pixels (e.g. macroblocks of 16×16 pixels in MPEG). Each block is predicted from a block of equal size in the reference frame. The blocks are not transformed in any way apart from being shifted to the position of the predicted block. This shift is represented by a *motion vector*.

To exploit the redundancy between neighboring block vectors, (e.g. for a single moving object covered by multiple blocks) it is common to encode only the difference between the current and previous motion vector in the bit-stream. The result of this differencing process is mathematically equivalent to a global motion compensation capable of panning. Further down the encoding pipeline, an entropy coder will take advantage of the resulting statistical distribution of the motion vectors around the zero vector to reduce the output size.

It is possible to shift a block by a non-integer number of pixels, which is called *sub-pixel precision*. The in-between pixels are generated by interpolating neighboring pixels. Commonly, half-pixel or quarter pixel precision (Qpel, used by H.264 and MPEG-4/ASP) is used. The computational expense of sub-pixel precision is much higher due to the extra processing required for interpolation and on the encoder side, a much greater number of potential source blocks to be evaluated.

The main disadvantage of block motion compensation is that it introduces discontinuities at the block borders (blocking artifacts). These artifacts appear in the form of sharp horizontal and vertical edges which are easily spotted by the human eye and produce false edges and ringing effects (large coefficients in high frequency sub-bands) due to quantization of coefficients of the Fourier-related transform used for transform coding of the residual frames[2]

Block motion compensation divides up the *current* frame into non-overlapping blocks, and the motion compensation vector tells where those blocks come *from* (a common misconception is that the *previous frame* is divided up into non-overlapping blocks, and the motion compensation vectors tell where those blocks move *to*). The source blocks typically overlap in the source frame. Some video compression algorithms assemble the current frame out of pieces of several different previously-transmitted frames.

Frames can also be predicted from future frames. The future frames then need to be encoded before the predicted frames and thus, the encoding order does not necessarily match the real frame order. Such frames are usually predicted from two directions, i.e. from the I- or P-frames that immediately precede or follow the predicted frame. These bidirectionally predicted frames are called *B-frames*. A coding scheme could, for instance, be IBBPBBPBBPBB.

Further, the use of triangular tiles has also been proposed for motion compensation. Under this scheme, the frame is tiled with triangles, and the next frame is generated by performing an affine transformation on these triangles.[3] Only the affine transformations are recorded/transmitted. This is capable of dealing with zooming, rotation, translation etc.

Variable block-size motion compensation

Variable block-size motion compensation (VBSMC) is the use of BMC with the ability for the encoder to dynamically select the size of the blocks. When coding video, the use of larger blocks can reduce the number of bits needed to represent the motion vectors, while the use of smaller blocks can result in a smaller amount of prediction residual information to encode. Older designs such as H.261 and MPEG-1 video typically use a fixed block size, while newer ones such as H.263, MPEG-4 Part 2, H.264/MPEG-4 AVC, and VC-1 give the encoder the ability to dynamically choose what block size will be used to represent the motion.

Overlapped block motion compensation

Overlapped block motion compensation (OBMC) is a good solution to these problems because it not only increases prediction accuracy but also avoids blocking artifacts. When using OBMC, blocks are typically twice as big in each dimension and overlap quadrant-wise with all 8 neighbouring blocks. Thus, each pixel belongs to 4 blocks. In such a scheme, there are 4 predictions for each pixel which are summed up to a weighted mean. For this purpose, blocks are associated with a window function that has the property that the sum of 4 overlapped windows is equal to 1 everywhere.

Studies of methods for reducing the complexity of OBMC have shown that the contribution to the window function is smallest for the diagonally-adjacent block. Reducing the weight for this contribution to zero and increasing the other weights by an equal amount leads to a substantial reduction in complexity without a large penalty in quality. In such a scheme, each pixel then belongs to 3 blocks rather than 4, and rather than using 8 neighboring blocks, only 4 are used for each block to be compensated. Such a scheme is found in the H.263 Annex F Advanced Prediction mode

Quarter Pixel (QPel) and Half Pixel motion compensation

In motion compensation, quarter or half samples are actually interpolated sub-samples caused by fractional motion vectors. Based on the vectors and full-samples, the sub-samples can be calculated by using bicubic or bilinear 2-D filtering. See subclause 8.4.2.2 "Fractional sample interpolation process" of the H.264 standard.

3D image coding techniques

Motion compensation is utilized in Stereoscopic Video Coding

In video, *time* is often considered as the third dimension. Still image coding techniques can be expanded to an extra dimension.

JPEG2000 uses wavelets, and these can also be used to encode motion without gaps between blocks in an adaptive way. Fractional pixel affine transformations lead to bleeding between adjacent pixels. If no higher internal resolution is used the delta images mostly fight against the image smearing out. The delta image can also be encoded as wavelets, so that the borders of the adaptive blocks match.

2D+Delta Encoding techniques utilize H.264 and MPEG-2 compatible coding and can use motion compensation to compress between stereoscopic images.

See also

- Motion estimation
- Image stabilization
- Inter frame
- HDTV blur
- Television standards conversion
- VidFIRE
- X-Video Motion Compensation

Applications

- video compression
- change of framerate for playback of 24 frames per second movies on 60 Hz LCDs or 100 Hz interlaced cathode ray tubes

References

1. berkeley.edu - Why do some people hate B-pictures? (https://web.archive.org/web/20090220062554/http://bmrc.berkeley.edu/research/mpeg/faq/mpeg2-v38/faq_v38.html#tag40)
2. Zeng, Kai, et al. "Characterizing perceptual artifacts in compressed video streams." IS&T/SPIE Electronic Imaging. International Society for Optics and Photonics, 2014.
3. Aizawa, Kiyoharu, and Thomas S. Huang. "Model-based image coding advanced video coding techniques for very low bit-rate applications." Proceedings of the IEEE 83.2 (1995): 259-271.

Garnham, N. W., Motion Compensated Video Coding, University of Nottingham PhD Thesis, October 1995, OCLC 59633188 (https://www.worldcat.org/oclc/59633188).

External links

- Temporal Rate Conversion (http://msdn.microsoft.com/en-us/windows/hardware/gg463407) - article giving an overview of motion compensation techniques.
- A New FFT Architecture and Chip Design for Motion Compensation based on Phase Correlation (http://portal.acm.org/citation.cfm?id=784892.784978)
- DCT and DFT coefficients are related by simple factors (http://vision.arc.nasa.gov/publications/mathjournal94.pdf)
- DCT better than DFT also for video (http://actapress.com/PaperInfo.aspx?PaperID=26756&reason=500)
- John Wiseman, An Introduction to MPEG Video Compression (http://old.siggraph.org/education/materials/HyperGraph/video/mpeg/)
- DCT and motion compensation (http://ieeexplore.ieee.org/Xplore/login.jsp?url=/iel5/76/18597/00856453.pdf?arnumber=856453)
- Compatibility between DCT, motion compensation and other methods (http://www.hindawi.com/GetArticle.aspx?doi=10.1155/S1110865701000245)

Categories: Film and video technology | Video compression