

Automatically Detecting Camera Motion from MPEG-1 Encoded Video

Stephen Donnelly, Alan F. Smeaton, Catherine Berrut
Seán Marlow, Noel Murphy and Noel O'Connor

Centre for Digital Video Processing
Dublin City University, Dublin 9, Ireland
asmeaton@compapp.dcu.ie

Keywords: Digital Video; MPEG-1; Camera Motion

We have developed, implemented and tested a technique for detecting camera motion directly from MPEG-1 encoded video that is based on processing the motion vectors in the bitstream. Detecting camera motion is equivalent to reverse engineering, from the video, in which direction the camera is turning during a shot. Camera motion parameters include the direction and magnitude of the camera turn, whether the motion is tracking, panning, booming or tilting and whether zoom-in or -out is being used. Camera motion forms a useful indexing feature for video in some applications such as logging news footage. In the work reported here, we simply detect the direction of camera motion and other aspects are targeted as future work.

Our system detects motion in real time as video is decoded and played based on the following process:

1. A sequence parser extracts frames from an MPEG-1 video stream in the order in which the frames are to be played.
2. We then extract macroblocks and their associated motion vectors; in MPEG-1 encoding there are 396 (22x18) such blocks. MPEG encoding is based on calculating the displacement of macroblocks between adjacent frames using a simple matching criterion, and representing this displacement as a motion vector. In our work we use an encoder from Optibase which has an I-frame (i.e. a video frame encoded as a still image, implying no motion vectors) every 12th frame. We extract motion vectors between I-frames and subsequent B-frames (i.e. video frames predictively encoded from previous and/or future frames). At a frame rate of 25 Hz, this means we sample camera motion about every 0.5 seconds.
3. Motion vector filtering is then applied to eliminate any rogue vectors. Rogue vectors are generally unavoidable since they occur as a consequence of the fact that macroblock displacement computation is based on matching blocks of pixels as opposed to calculating real motion. We categorise each non-zero motion vector into one of 90°, 180°, 270° and 360° bins as these represent the most common camera motion directions, certainly the ones of interest for indexing, and we use the fullest bin over a set threshold as the camera motion.

Figure 1 shows the motion vectors for an I- to B-frame transition where the motion vectors are predominantly pointing to the left, indicating that the camera has been panning to the left (note the rogue vectors and areas of no motion).

To date we have tested our approach on 3 clips of video: a soccer match, an episode of a soap opera, and a clip from a Formula 1 Grand Prix where we manually measured the actual camera motion and compared against the estimated motion. Of the frame transitions in this 3 minutes of footage we are correct for 85% of the time.

Our future work will involve a larger collection of video, a more sophisticated evaluation measure, the inclusion of motion vector *magnitude* as well as direction, and the development and

evaluation of more sophisticated approaches to camera motion detection and classification.

