Master VICO – Lab on Video Coding

Motion Estimation (ME)

Goals:

We want to study and compare the performances of different motion estimation methods based on block matching.

Introduction:

We are going to use for the lab *VCDemo* from TUDelft, and particularly its module *Motion Estimation* (ME). To download *VCDemo*:
http://insy.ewi.tudelft.nl/content/image-and-video-compression-learning-tool-vcdemo

This module implements several motion estimation methods, all are Block Matching based and use the frames luminance component. For instance you can choose between several approaches and sets:

- simple resolution / multi-resolution (hierarchical with 2 levels)
- full search / One-at-a-time / N-Step
- Windows search size

After video processing, a set of results and data are given:

- block motion vector
- motion compensation error images
- inter-image differences.

Note that the displacement vectors and prediction errors are used, in an appropriate form, by the MPEG-1/2 standards or the MPEG-4 / AVC standard (also called MPEG-4 Part 10 or H. 264).

To do:

Search on the Net what are the differences between these motion estimation methods (block-matching, full search, N-step, one-a-time, mono and multi-resolution approaches).

A – Motion estimation by block-matching, mono-resolution

The used video is the raw video (YUV format) *Vectra*, this sequence counts 21 frames.

A1. Full Search

For an 8x8 block size, and by considering different sizes of the search window (e.g: maximum displacements set to 3, 7, 15, 24, or 31 pixels), analyze (visually and quantitatively) the performance and the quality of the motion estimation.

What is then the best size for the search window?

A2. Blocks size

We use the best size, found previously, for the search window.

We play now with 4 block sizes (2x2, 4x4, 8x8, 16x16), and test them on the complete video sequence.

What is the best block size on this sequence?

If we take into account the coding cost rate of the motion vectors, what is then the optimal block size for this sequence?

A3. N-step search

To reduce the computational complexity, we use the N-step search method.

Considering 8x8 block size, and the previous search window size, we use the One-at-a-time method. Analyze (visually and quantitatively) the performance and the quality of the motion estimation.

Do the same with the N-step method for N=2, 3, 4 and 5 successively. Analyze (visually and quantitatively) the performance and the quality of the motion estimation.

Conclude on these 3 types of motion estimation.

B- Hierarchical block-matching

To reduce the computational cost, a hierarchical (multi-resolution) method is used.

B1. Multi-resolution estimation with 2 decomposition levels

The block size is 8x8. Analyze the ME performances for these different search windows sizes: 3-1, 7-3, and 15-7.

Compare with the mono-resolution full-search approach results (using their corresponding best search window sizes).

B2. Multi-resolution estimation with 3 decomposition levels

The block size is 8x8. Analyze the ME performances for these different search windows sizes: 7-3-1, and 15-7-3.

Compare with the results obtained with the 2 previous ME methods (hierarchical with 2 levels, mono-resolution).

Conclude on these types of motion estimation.

Estimate the computation cost of these 2 methods (for the same final search window size):

- mono-resolution
- hierarchical with 3 decomposition levels

Conclude on this aspect (quality of the ME / computing times).
