EUSIPCO 2005
Antalya
September 2005

A JOINT MOTION COMPUTATION AND SEGMENTATION ALGORITHM FOR VIDEO CODING

Sylvain Boltz, Eric Debreuve, Michel Barlaud
Laboratoire I3S, University of Nice-Sophia Antipolis / CNRS, FRANCE

Objectives

Improve temporal wavelet filtering for video coding

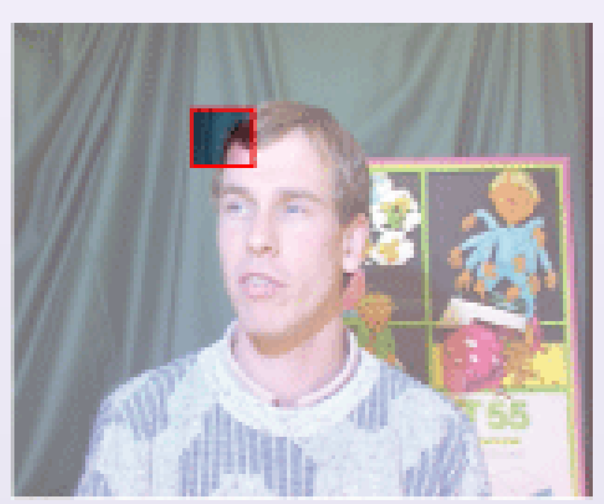
One method : divide the blocks into 4 smaller blocks

Proposed method : divide the blocks into 2 regions by motion segmentation

Minimize the energy of the HF subbands

Cleaner HF subbands, less flutter and blocking artifacts

I.Blocks versus Regions



Macroblock subdivision:

QuadTree: macroblock with high prediction error divided into smaller blocks

Puzzle: macroblock with high prediction error divided into 2 regions

This method should perform well on object borders

Transmission Cost per block:

QuadTree: 4 motion vectors

Puzzle: 2 motions vectors + 4 control points

Example : motion vectors at 1/8 pixel precision and control points at pixel precision:

64bits/macroblock in both cases

II. Joint motion segmentation and computation

Criterion definition:

Defined from the classical optical flow equation

$$(I(m,i) - I(m+v,i+1))^2 = 0$$

Energy

$$\begin{cases} J(\Omega) = \int_{\Omega} (I(m,i) - I(m+v(\Omega),i+1))^2 dm \\ v(\Omega) = \arg\min_{v} \int_{\Omega} (I(m,i) - I(m+v(\Omega),i+1))^2 dm \end{cases}$$

Bidirectional extension (forward and backward frame)

$$k(m,v) = (I(m,i) - I(m+v,i+1))^2 + (I(m,i) - I(m-v,i-1))^2$$

$$\begin{cases} J(\Omega) = \int_{\Omega} k(m,v(\Omega))dm \\ v(\Omega) = \arg\min_{v} \int_{\Omega} k(m,v)dm \end{cases}$$

Alternate minimization algorithm

Segmentation

Region competition: Ω unknown, motion fixed

$$J(\Omega) = \int_{\Omega} k(m,v(\Omega))dm + \int_{\bar{\Omega}} k(m,v(\bar{\Omega}))dm + \int_{\partial\Omega} \beta\, dt$$

Region dependant terms $v_1 = v(\Omega)$, $v_2 = v(\bar{\Omega})$, $\Gamma = \partial\Omega$

Dynamic scheme, evolution parameter τ $\quad J(\Omega(\tau)) = J(\tau)$

Shape gradient:

$$J'(\tau) = \int_{\Gamma(\tau)} k(m,v_2) - k(m,v_1)(V.N)ds + \int_{\Gamma(\tau)} (-\beta.\kappa)(V.N)ds$$

No additional terms: J and v are the "*same*" functional

Convergence to a minimum when $J'(\tau) = 0$

Evolution equation of the spline control points:

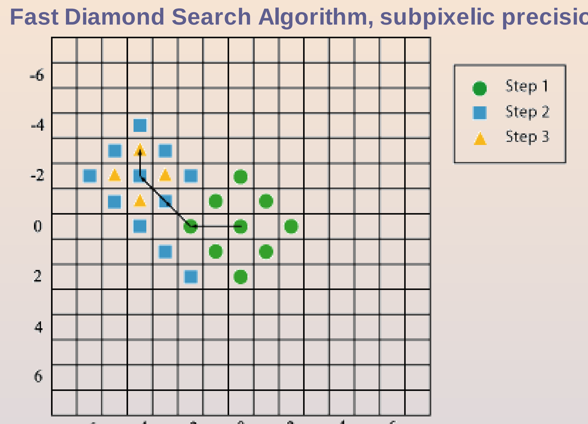
$$V = \frac{\partial\Gamma}{\partial\tau} = (k(m,v_1) - k(m,v_2) + \beta.\kappa).\vec{N}$$

Motion Estimation

Region matching

Block matching algorithm with submask

Fast Diamond Search Algorithm, subpixelic precision



Matching criterionl

$$v(\Omega) = \arg\min_{v} \sum_{x\in\Omega} k(m,v)$$

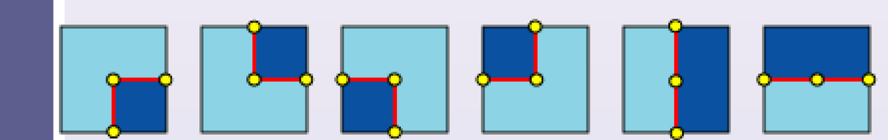
Occlusion management:

Weighting of the backward and forward frames

$$k(m,v) = c_f \times (I(m,i) - I(m+v,i+1))^2 + c_b \times (I(m,i) - I(m-v,i-1))^2$$

III. Implementation and results

Selection and block initialization



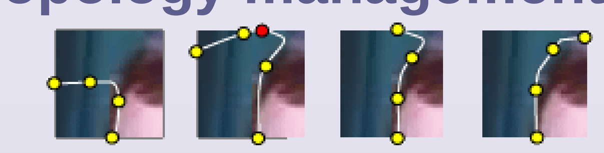
-Division of a macroblock into 4 smaller blocks

-Estimation of their motion vectors

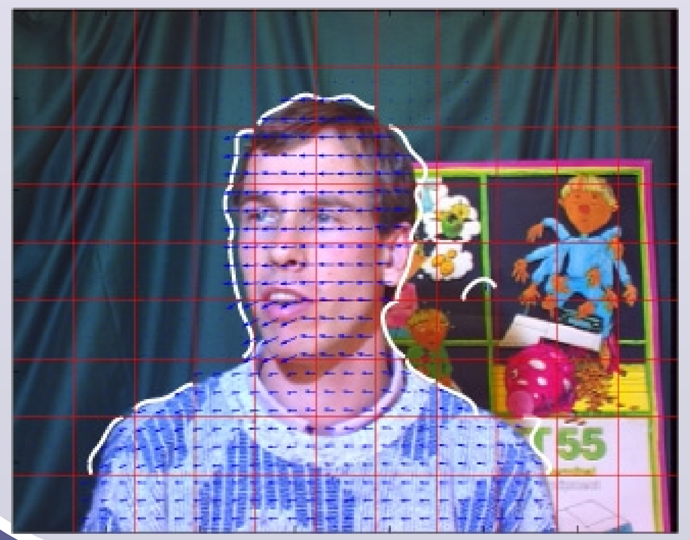
-Thresholding of the difference between these vectors

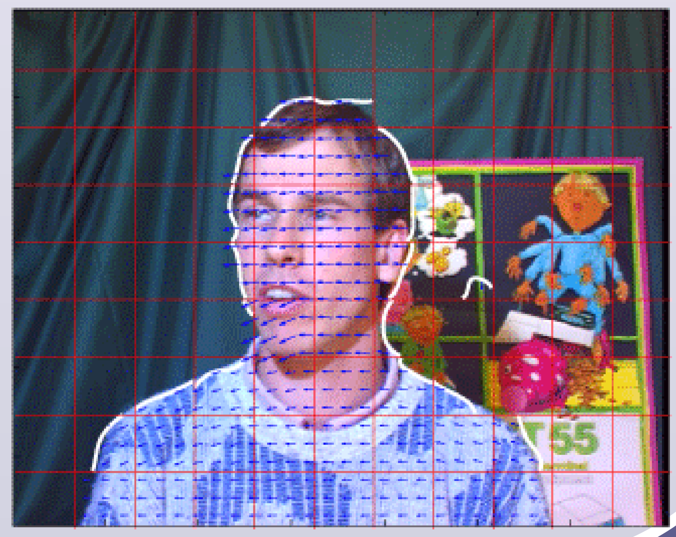
-> Initialization of the segmentation of the macroblock or rejection the macroblock

Topology management



Results with and without occlusion management





IV. Performances and conclusion

Average on 8 frames of sequence "Erik"

Macroblock division	4 blocks	2 regions	Gain
Average prediction error per frame	59.9	39.1	33.90%
Average prediction error per frame with occlusion management	57.9	42.8	25.83%

Energy gain around 30% in the high frequency subband

Future works: Implementation of this method in a full video coder

[André et.al., "*Full occlusion management for wavelet-based video coding*", proc. EUSIPCO'05]