Better Computer Vision Under Video Compression, An Example Using Mean Shift Tracking

Dr Salman Aslam

Wing Commander, PAF
Associate Professor
Avionics Department
College of Aeronautical Engineering
PAF Academy Risalpur

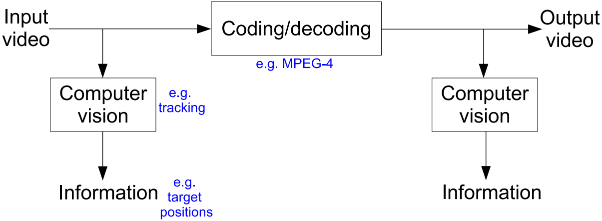
Outline

Introduction

goal

We want to run some computer vision algorithm (e.g. tracking) on compressed video.

- We would like the results of the algorithm (e.g. track positions) to be as close as possible to results generated if the algorithm were run on uncompressed video.



Introduction

possible solutions

1. Codec based solution
   - custom codec
   - standard codec
     - metadata support, e.g. MPEG-4
     - intelligently vary quantization parameter Q_p, or quantization matrices
2. Signal processing based solution
   - Vary input signal intelligently to make it robust to degradations in the encoding/decoding process

Introduction

our approach

Center for Signal & Image Processing

COMPUTATIONAL PERCEPTION LAB

Signal processing based solution

Calculus of Variations problem, with functional J

$$\min_{g} J(g) = \iint\limits_{D} \|f(I) - f(\hat{g}(I))\|_2 + \lambda \|s(I) - s(\hat{g}(I))\|_2 \,\mathrm{dxdy}$$

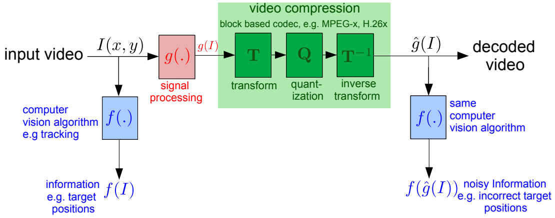
D is the domain of the image

$s(.)$ is an image similarity measure

Introduction

Our approach (cont.)

- Computer vision algorithm, $f(.)$:
  Mean shift tracking on hue space
- Preserving information, $g(.)$:
  Smooth out variations in hue space
  - A segmentation method can be used to manipulate the hue distribution
  - Modified stochastic gradient descent algorithm to find locally optimal parameters
  - Pick a local parameter space at random and optimize parameters locally



Mean shift filtering

notation

- N: number of data points
- D: number of dimensions
- $p(x)$: distribution of x
- R: small region containing K data points (a subset of all the data points in x)
- V: volume of R
- P: each point in x has probability P of falling in R, where

$$P = \int_R p(x)dx$$

- $\text{bin}(K|N, P)$: binomial distribution for K
- $k(u)$: unit square centered on origin, a Parzen window

$$k(u) = \begin{cases} 1, & |u_i| < 1/2 \qquad\qquad i = 1, \ldots, D \\ 0, & \text{otherwise} \end{cases}$$

Mean shift filtering

adaptive gradient ascent

$$p_k(x) = \frac{c_k}{Nh^D} \sum_{i=1}^{N} k\left(\|\frac{x-x_i}{h}\|^2\right)$$

c_k is a normalization constant
$k\left(\|\frac{x-x_i}{h}\|^2\right)$ is a kernel

computing the gradient,

$$\nabla p_k(x) = \frac{2c_k}{Nh^{D+2}} \sum_{i=1}^{N} (x - x_i) k'\left(\|\frac{x-x_i}{h}\|^2\right)$$

$$= \frac{-2c_k}{Nh^{D+2}} \left[\sum_{i=1}^{N} x_i k'\left(\|\frac{x-x_i}{h}\|^2\right) - x \sum_{i=1}^{N} k'\left(\|\frac{x-x_i}{h}\|^2\right) \right]$$

$$= \frac{-2c_k}{Nh^{D+2}} \sum_{i=1}^{N} k'\left(\|\frac{x-x_i}{h}\|^2\right) \underbrace{\left[\frac{\sum_{i=1}^{N} x_i k'\left(\|\frac{x-x_i}{h}\|^2\right)}{\sum_{i=1}^{N} k'\left(\|\frac{x-x_i}{h}\|^2\right)} - x \right]}_{\text{mean shift vector } m_{k'}(x)}$$

$$= \frac{-2c_k c_{k'}}{c_{k'} h^2 N h^D} \underbrace{\sum_{i=1}^{N} k'\left(\|\frac{x-x_i}{h}\|^2\right)}_{p_{k'}(x)} m_{k'}(x)$$

$$= \frac{2c}{h^2} p_{k'}(x) m_{k'}(x)$$

$$m_{k'}(x) = \frac{h^2}{2c} \frac{\nabla p_k}{p_{k'}(x)}$$

mean shift vector $m_{k'}(x)$ always points in direction of maximum increase in density

Mean shift filtering

simplification with uniform kernel

New target center, x_c and y_c are computed as,

$$M_{00} = \sum_x \sum_y I(x, y)$$

$$M_{10} = \sum_x \sum_y xI(x, y)$$

$$M_{01} = \sum_x \sum_y yI(x, y)$$

$$x_c = \frac{M_{10}}{M_{00}}$$

$$y_c = \frac{M_{01}}{M_{00}}$$

Mean shift segmentation

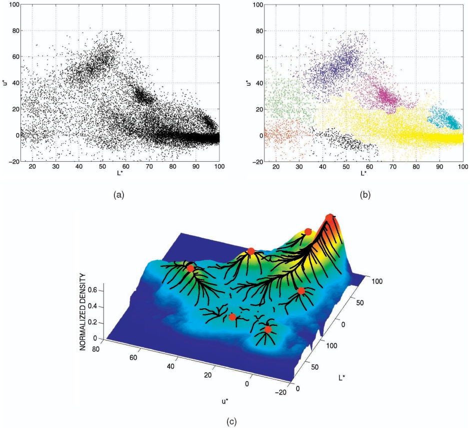
steps

1. Mean shift filtering
   - For a pixel, compute its mean shift vector
   - This mean shift vector ends at a mode of the density
   - Repeat for all pixels, now you have a bunch of modes
   - The set of all locations that converge to the same mode defines the *basin of attraction* of that mode
2. Merge basins of attraction
   - Merge (i.e. prune) modes that are close to each other, and corresponding basins of attraction

Mean shift segmentation

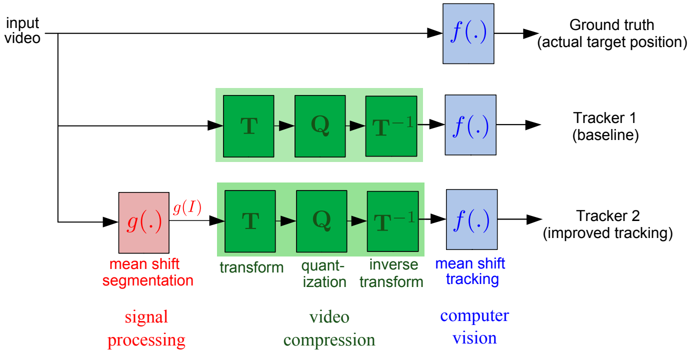
example

- 7 pruned nodes
- 7 corresponding basins of attraction (i.e. clusters, or segmented regions)



Comaniciu et. al., 2002

Experimental Setup



Results

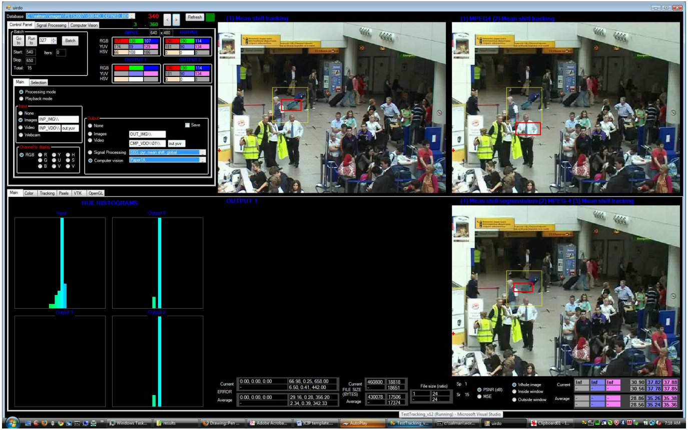
PETS2001

- person tracking
- sparse background
- target occluded by object (car) with similar color distribution

Results

PETS2007

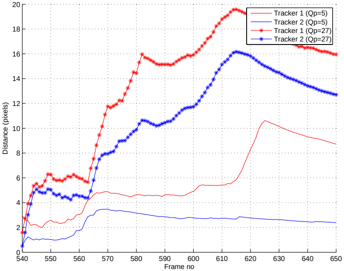
- object (bag) tracking
- dense background
- target occluded by object with similar color distribution



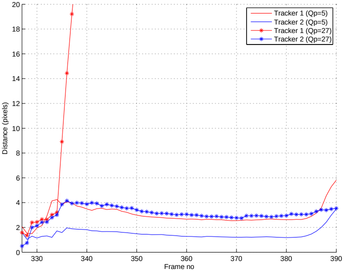
Results

tracking error

PETS2001



PETS2007



- Original tracker can lose track while improved tracker remains within an average of 5 pixels of target center
- Same bitrate
- PSNR drop for entire image is around 2 dB
- PSNR drop inside ROI is between 2 and 11 dB

Conclusions

1. It is possible to preserve information in a compressed video signal for a given application
   - In this work, we focused on tracking at low bit-rates
   - challenging scenes were used
   - person or a bag was tracked after being occluded by another object with similar color distribution
2. Tracking accuracy was improved in 95 % of the cases tested
   - average PSNR drop of 5dB inside the segmentation window for a given bitrate