MobileASL

Distortion-Complexity Optimization of the H.264/MPEG-4 AVC Encoder using the GBFOS Algorithm

Rahul Vanam, Eve Riskin,

Sheila Hemami and Richard Ladner

University of Washington,

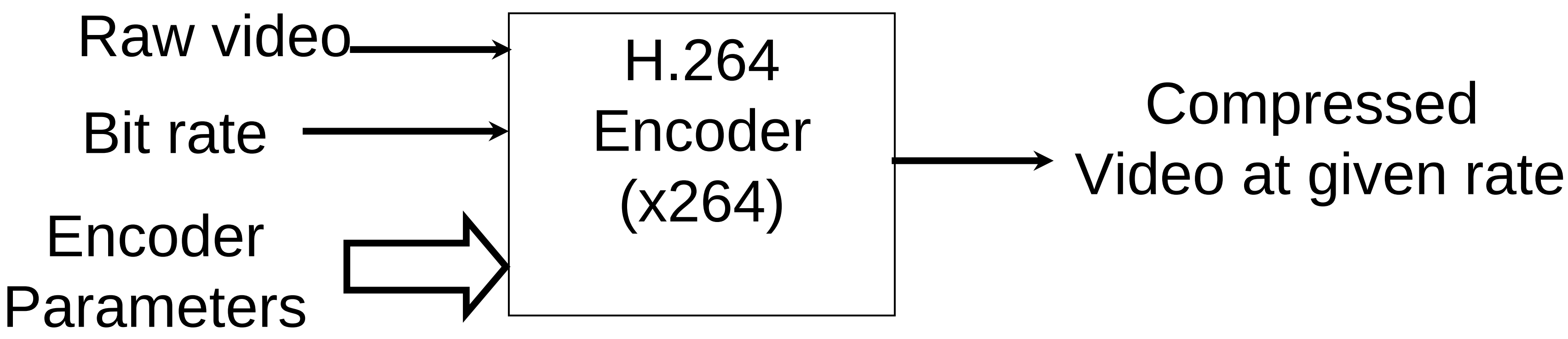
Cornell University

UNIVERSITY OF WASHINGTON

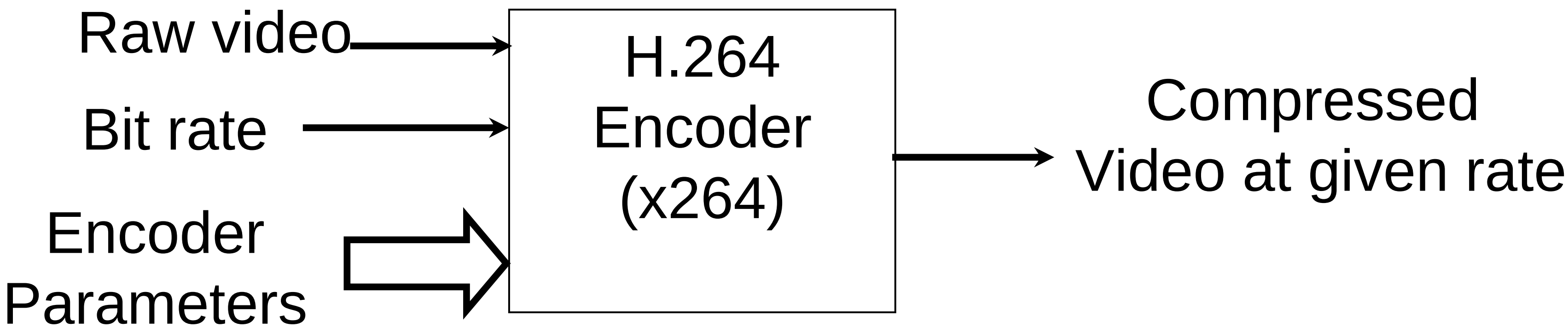
Cornell University

Data Compression Conference 2007

Problem


Raw video
Bit rate
Encoder
Parameters
H.264
Encoder
(x264)
Compressed
Video at given rate

Problem

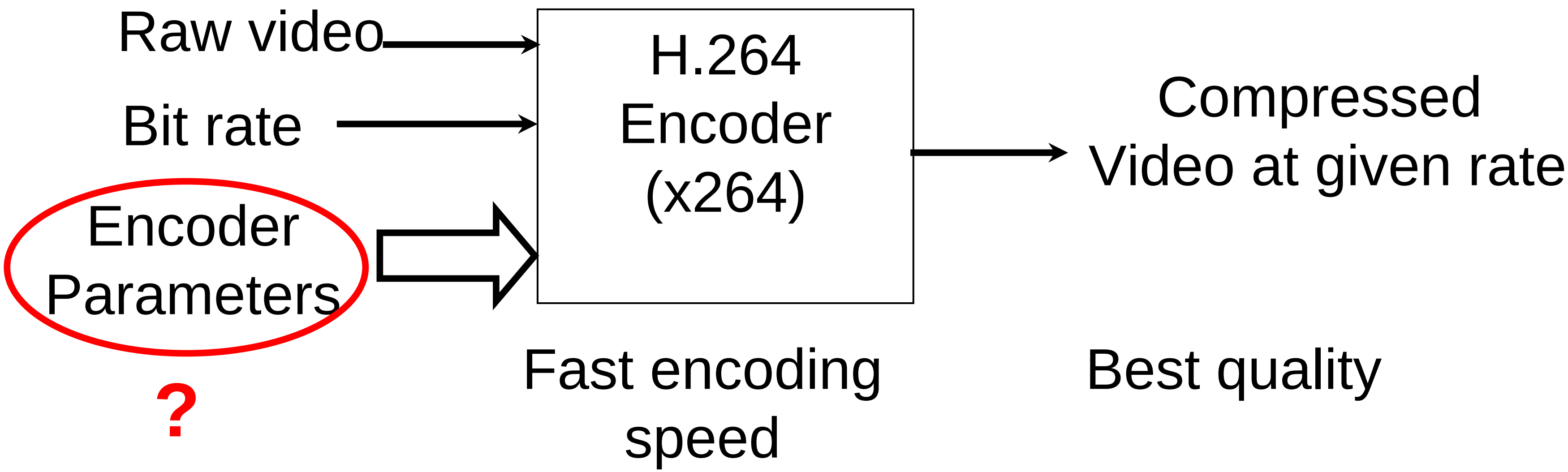


Real time application

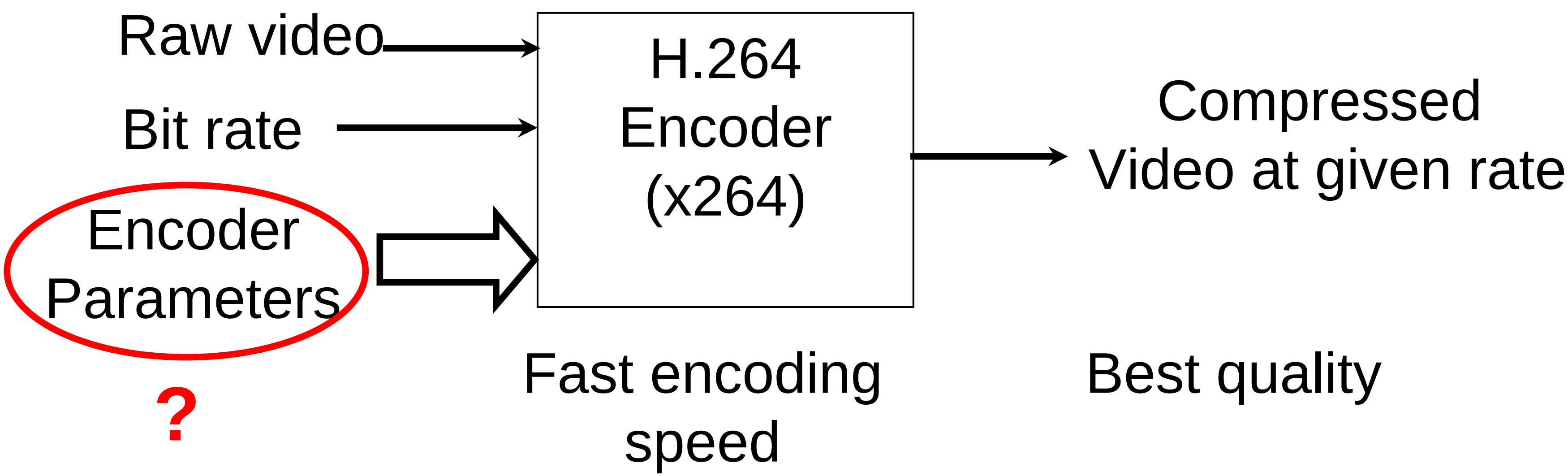
Fast encoding speed

Best quality

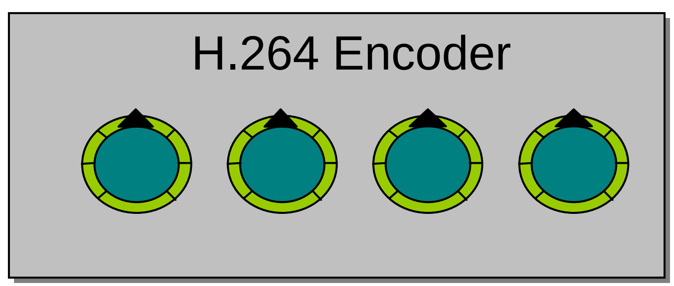
Problem


Raw video
Bit rate
Encoder Parameters
?
H.264 Encoder (x264)
Compressed Video at given rate
Fast encoding speed
Best quality

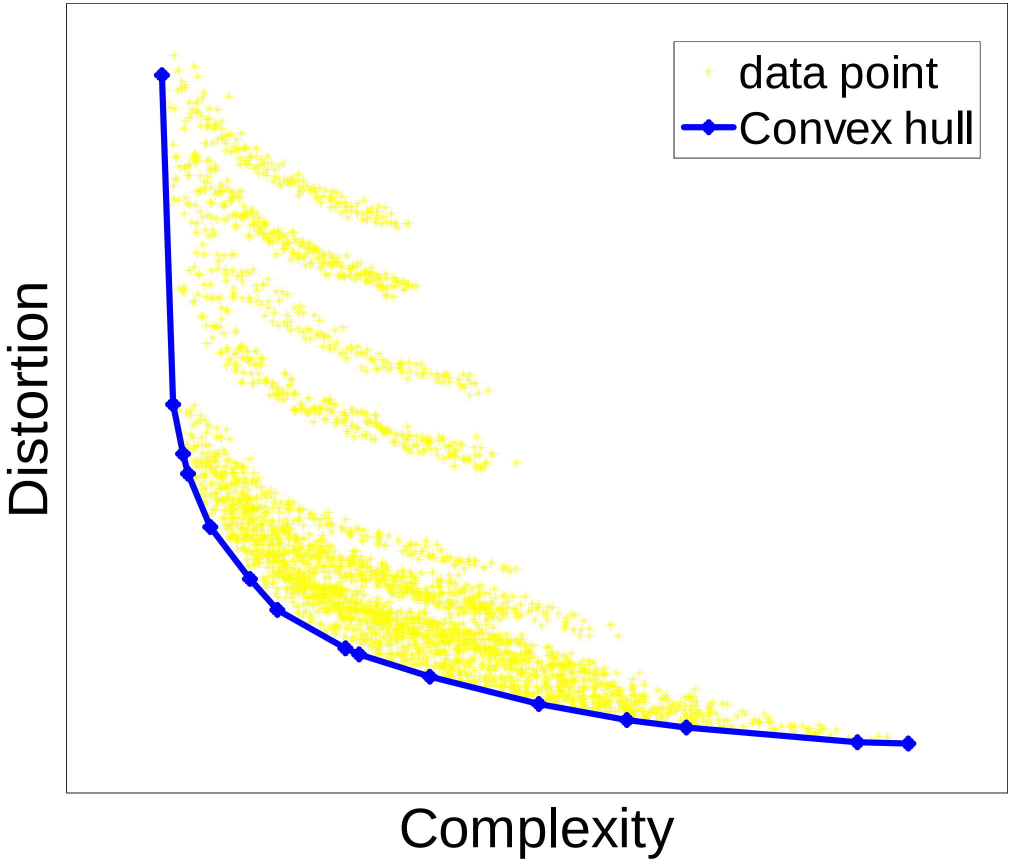
Problem



Parameters are similar to knobs. As a knob is moved from lower to higher setting: quality improves while speed suffers



Selecting the Encoder Parameters



Exhaustive search:

- Run the encoder for all possible settings.
- Obtain MSE vs. encode time plot
- Choose settings on the Distortion-Complexity convex hull
- Extremely time consuming!

Selecting Encoder Parameters

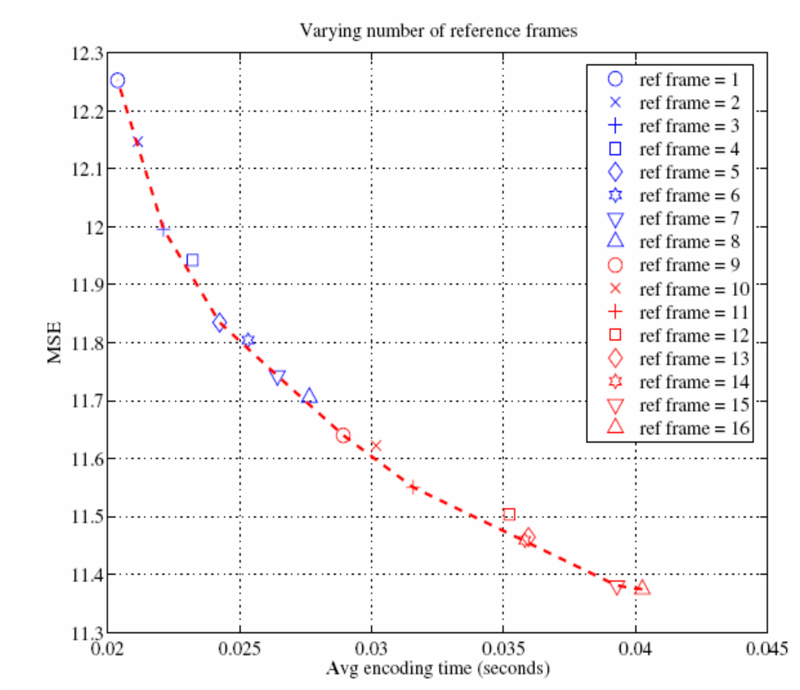
- GBFOS-basic and GBFOS-iterative algorithms select good encoder parameter settings with fewer encodings
- Based on the GBFOS (Generalized Breiman, Friedman, Olshen and Stone) algorithm
- ☆ GBFOS Assumption: Distortion and Complexity of parameters are additive/independent (untrue!)

Selecting Encoder Parameters

- Both algorithms are explained using an example:
  - Four parameter setting variables: number of reference frames, partition size, sub pixel motion estimation and quantization method
  - Other settings are constant

GBFOS-Basic Algorithm

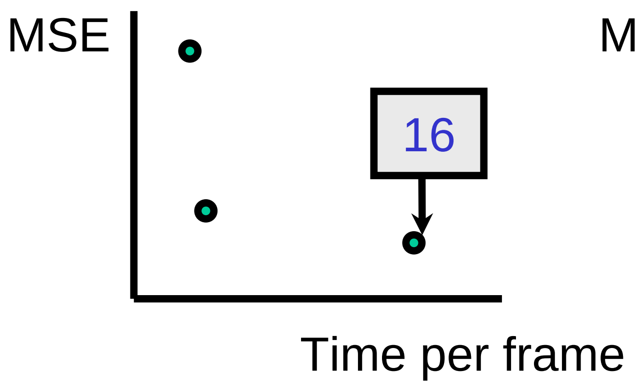
- Obtain D-C plot for each parameter by setting other parameters to their best PSNR option
- Choose convex hull points and corresponding parameter settings and slopes



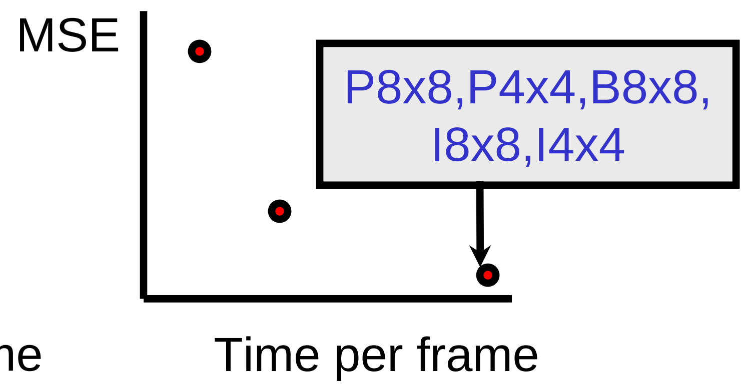
Distortion-complexity plot for Number of Reference frames

GBFOS-Basic Algorithm

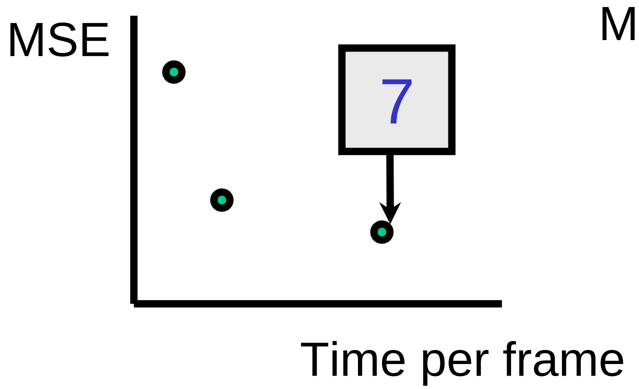
Reference frames



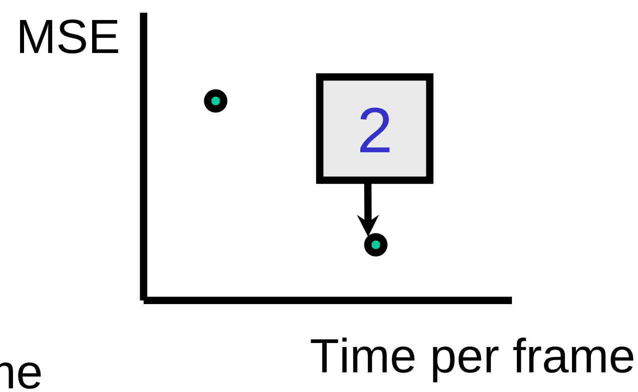
Partition sizes



Subme



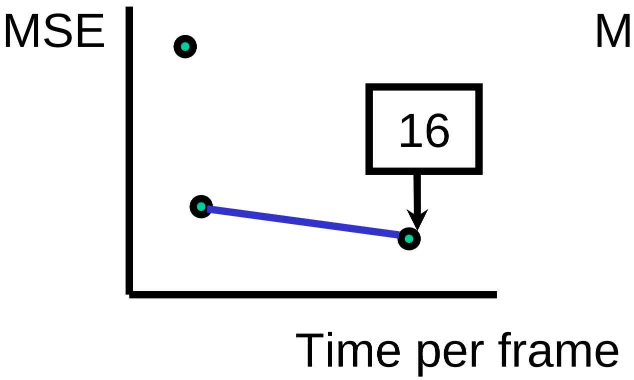
Trellis



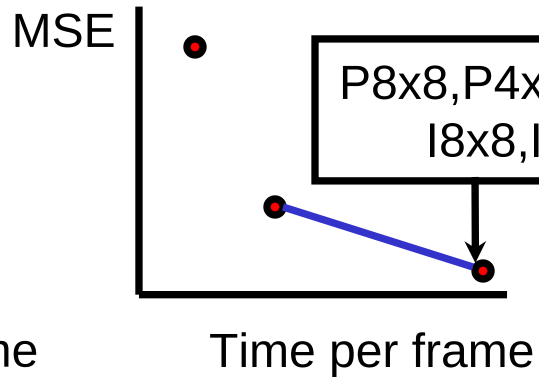
First parameter setting selected

GBFOS-Basic Algorithm

Reference frames

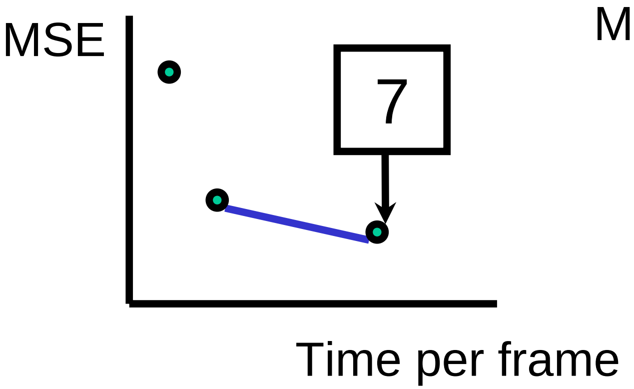


Partition sizes

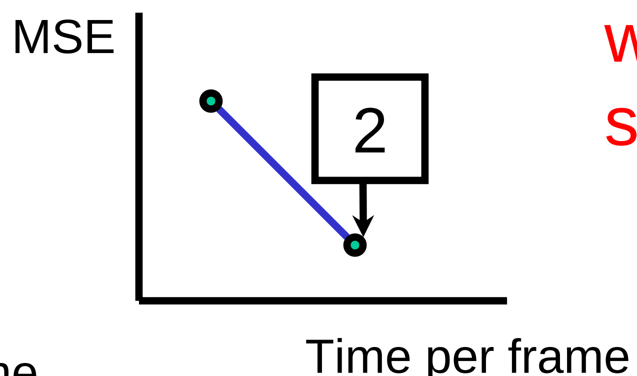


Compare slopes

Subme

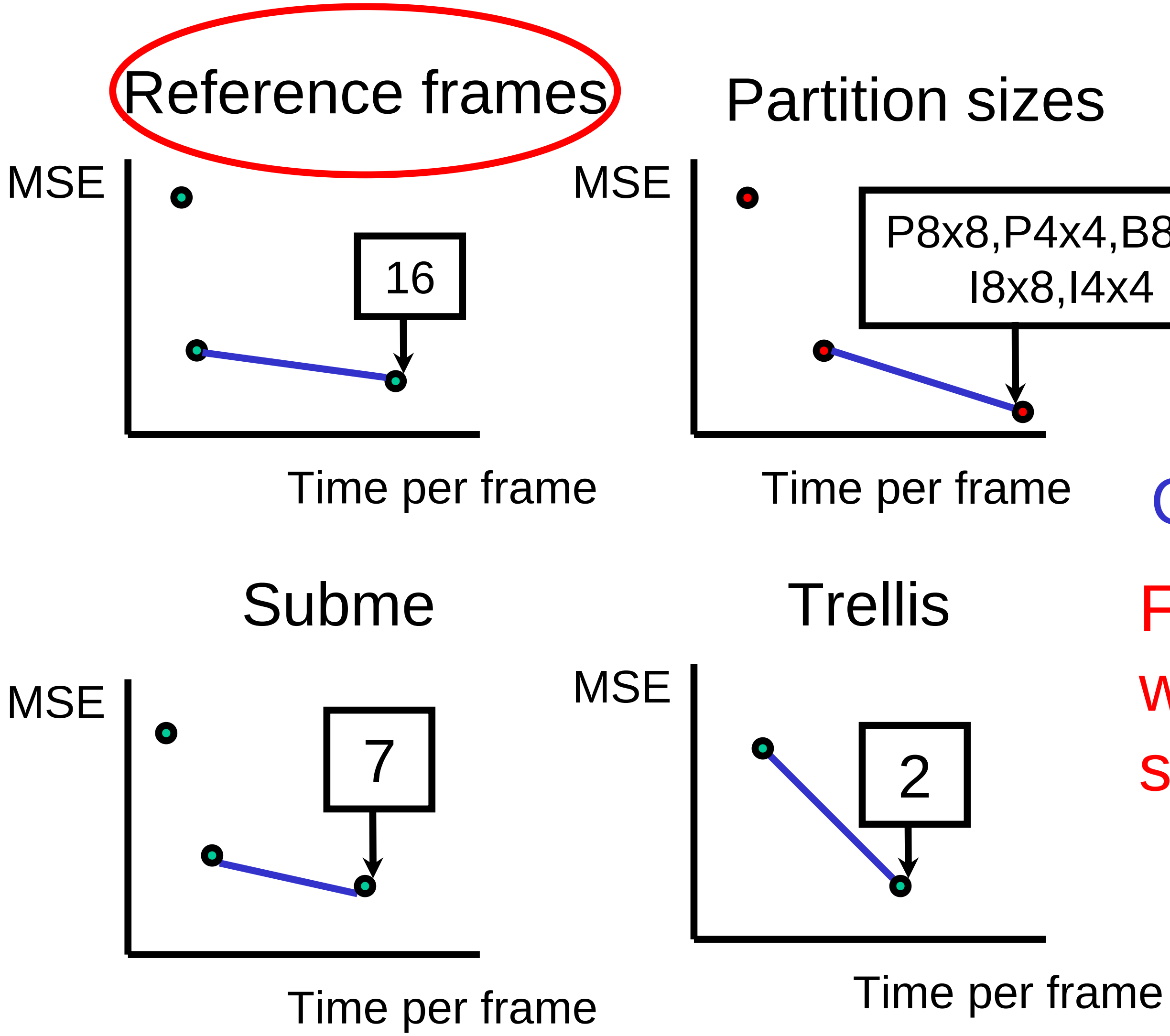


Trellis



Find the parameter with the least slope

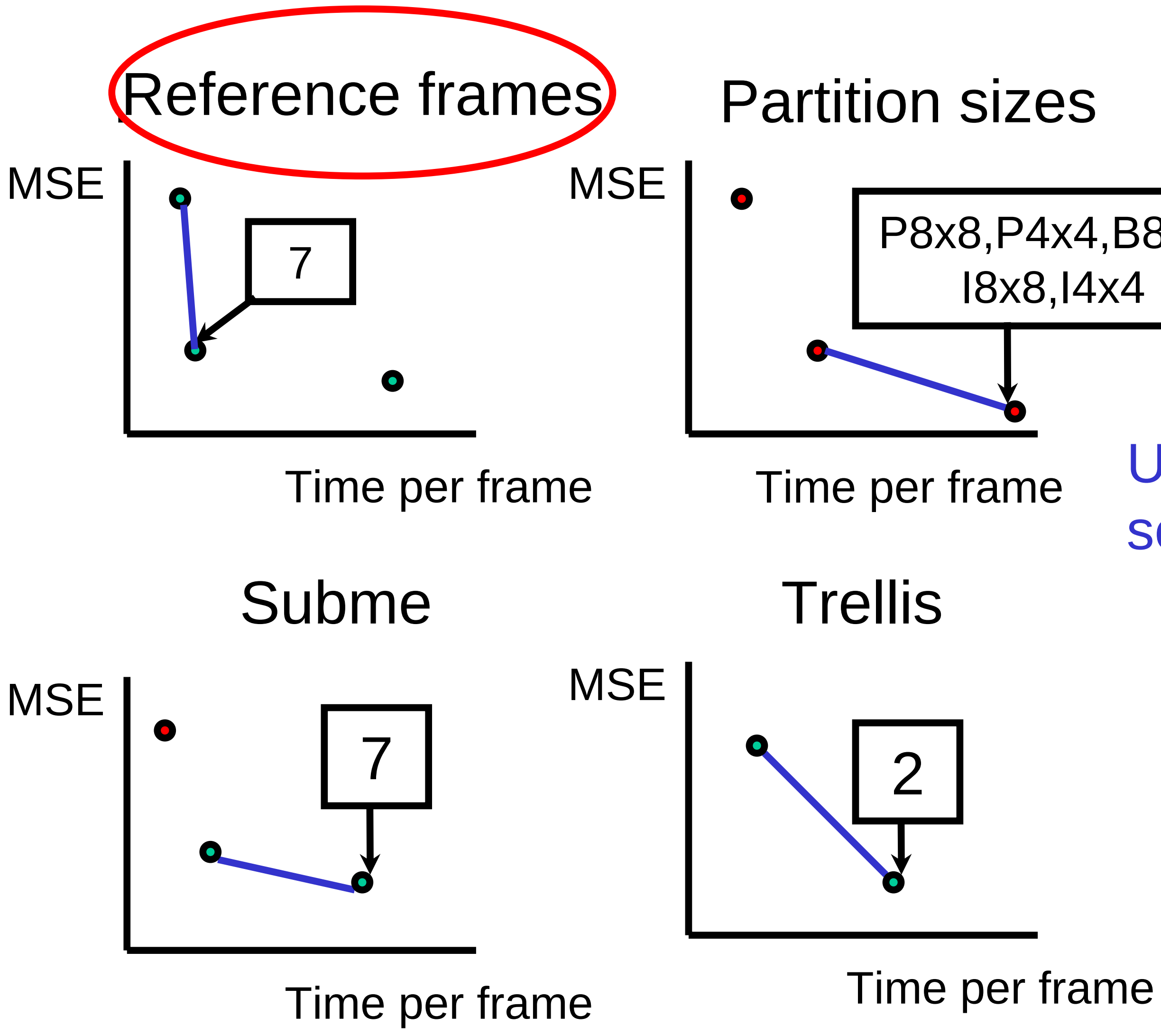
GBFOS-Basic Algorithm



Compare slopes

Find the parameter with the least slope

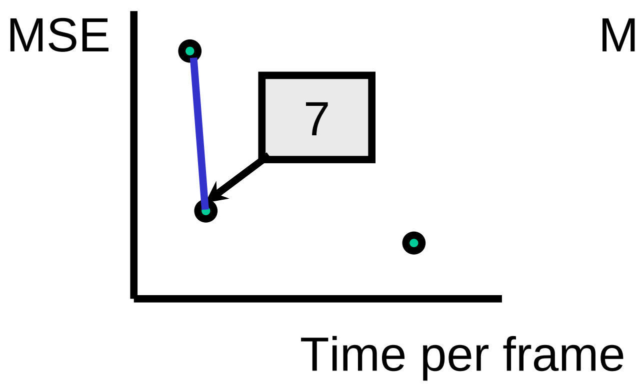
GBFOS-Basic Algorithm



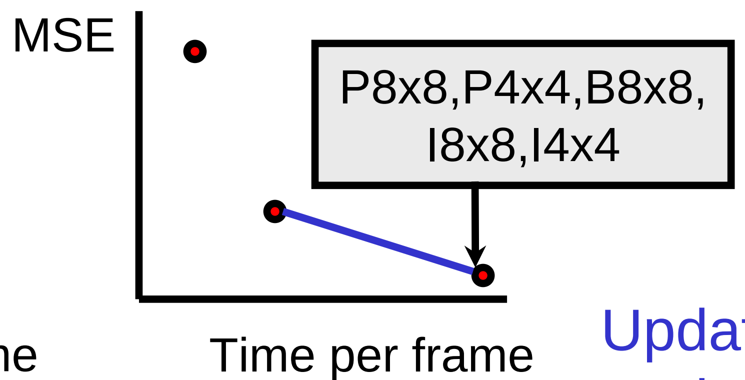
Update the slope and setting of Ref. frames

GBFOS-Basic Algorithm

Reference frames

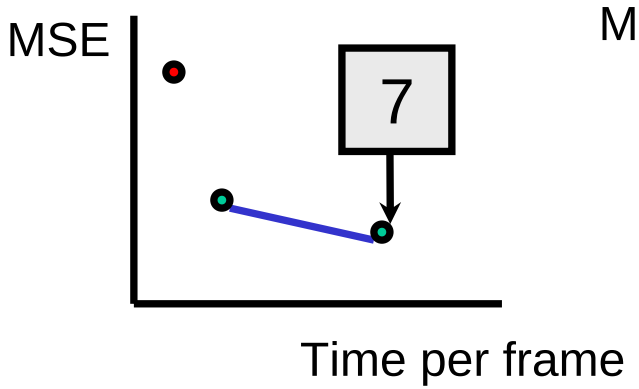


Partition sizes

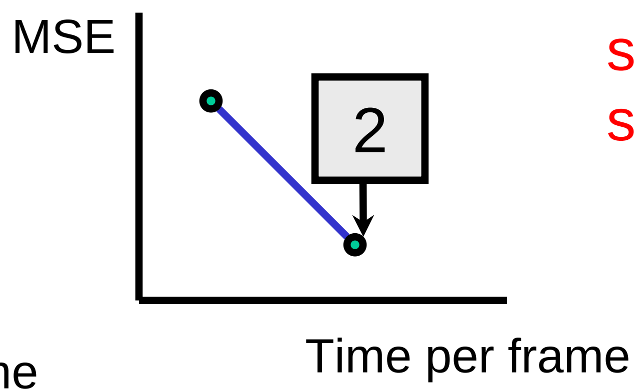


Update the slope and setting of Ref. frames

Subme

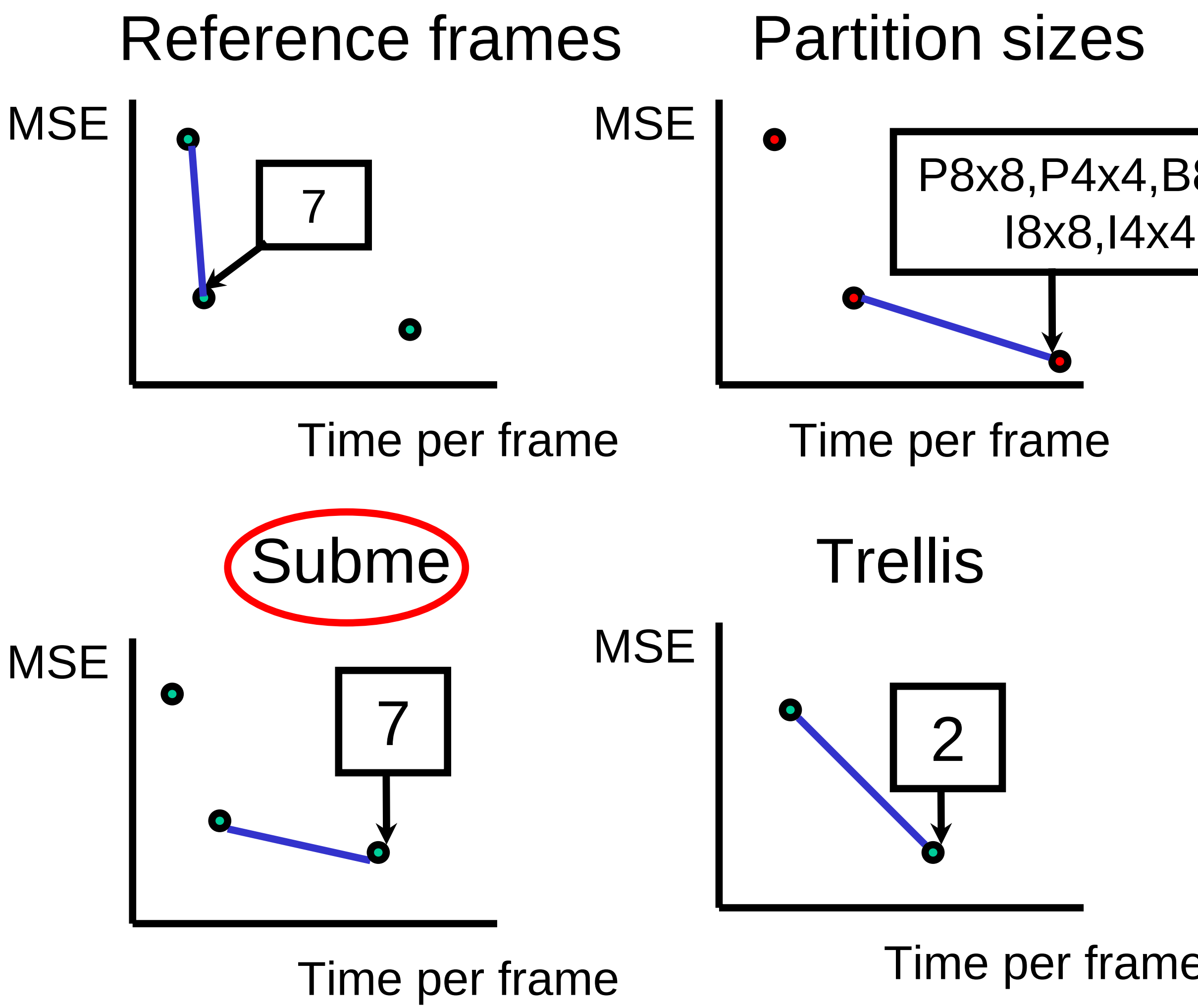


Trellis



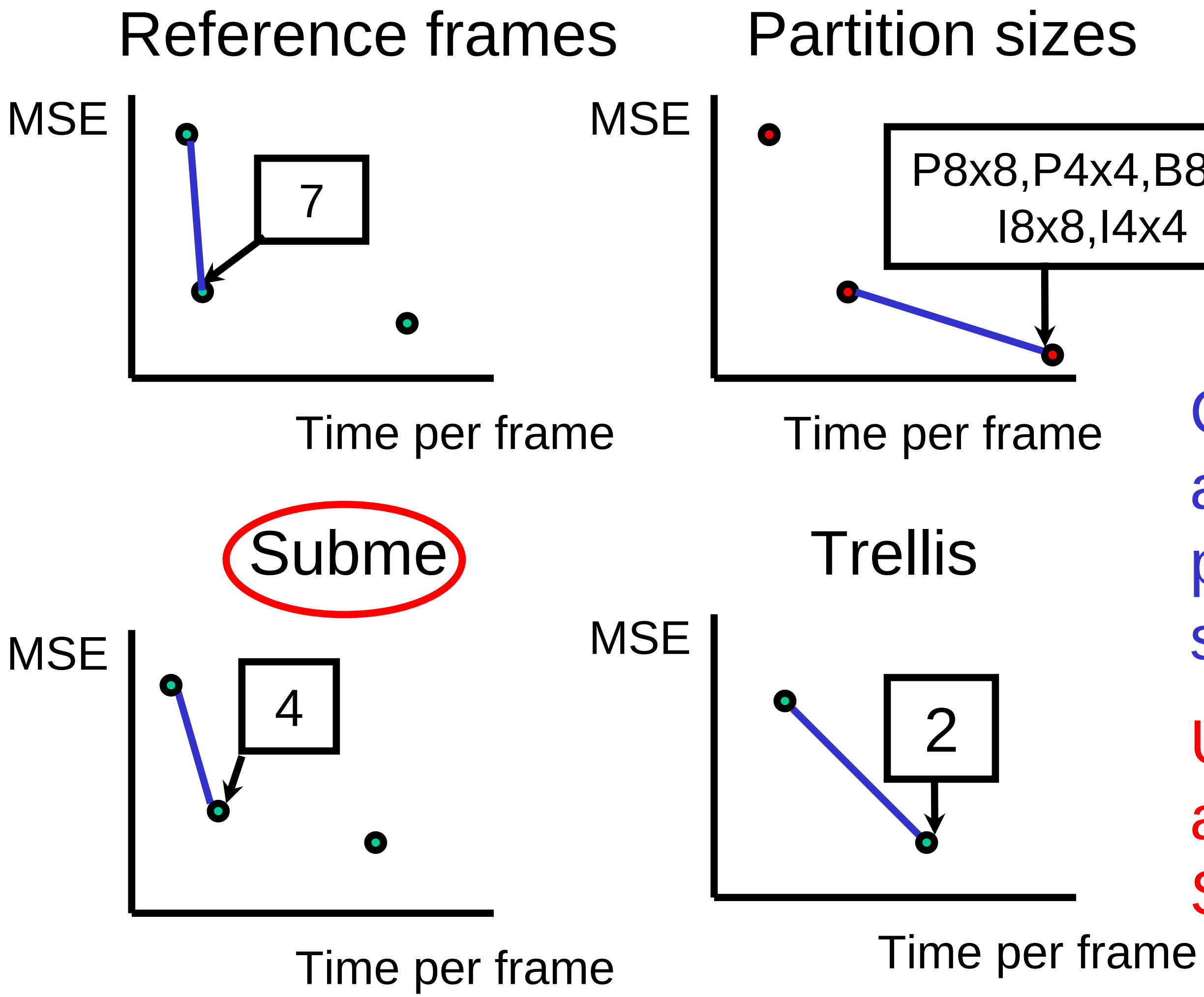
We now have the second parameter setting

GBFOS-Basic Algorithm



Compare the slopes again and find the parameter with least slope

GBFOS-Basic Algorithm

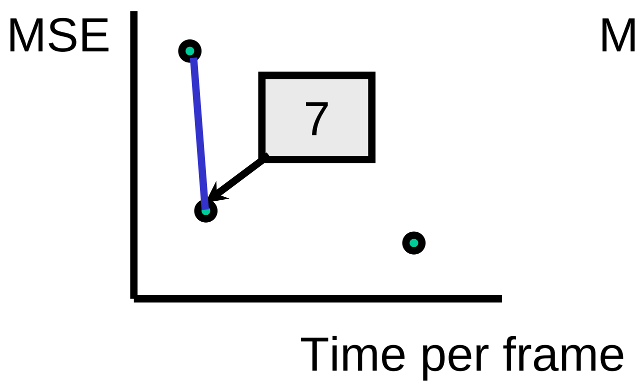


Compare the slopes again and find the parameter with least slope

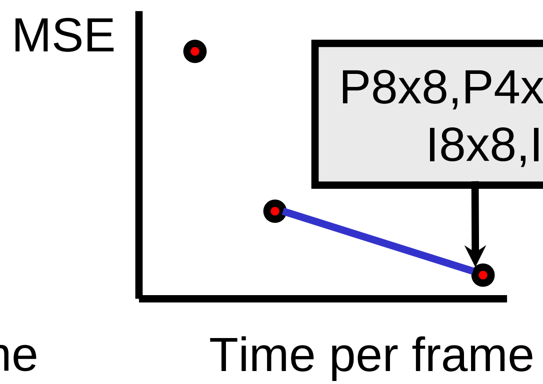
Update the slope and setting of Subme

GBFOS-Basic Algorithm

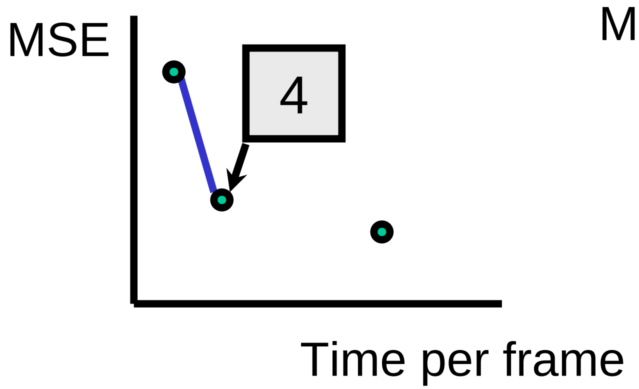
Reference frames



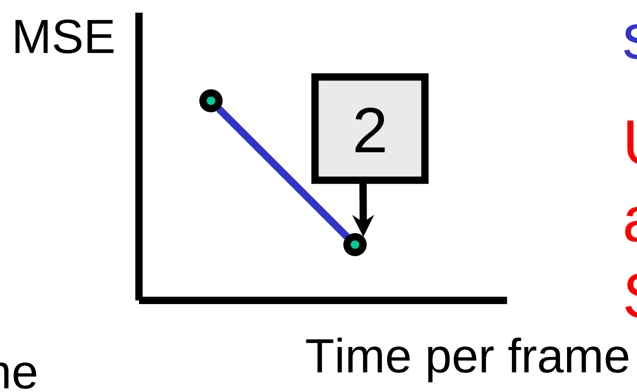
Partition sizes



Submme



Trellis

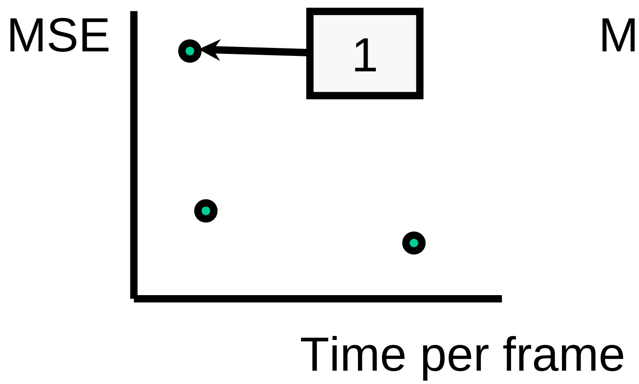


Compare the slopes again and find the parameter with least slope

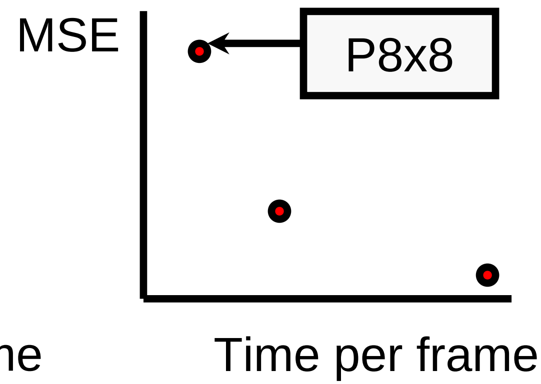
Update the slope and setting of Submme

GBFOS-Basic Algorithm

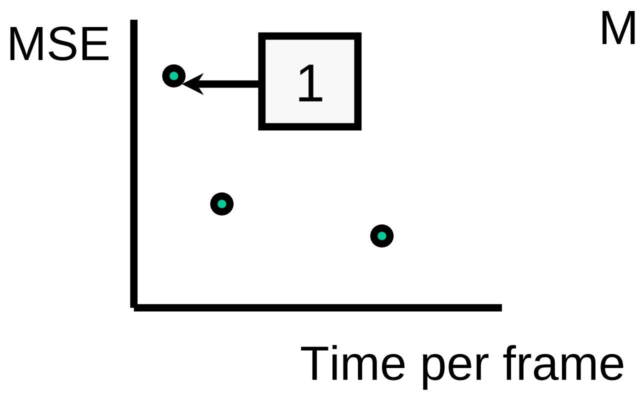
Reference frames



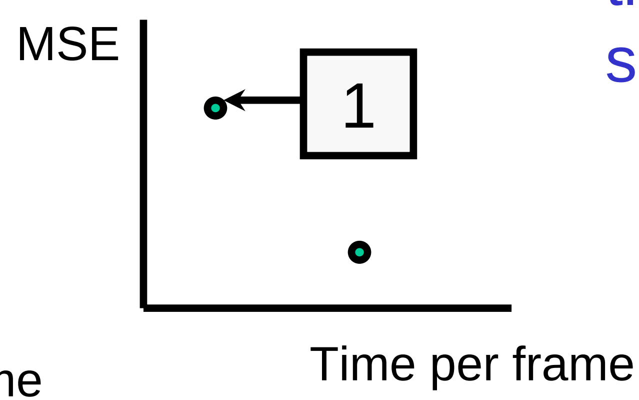
Partition sizes



Subme



Trellis

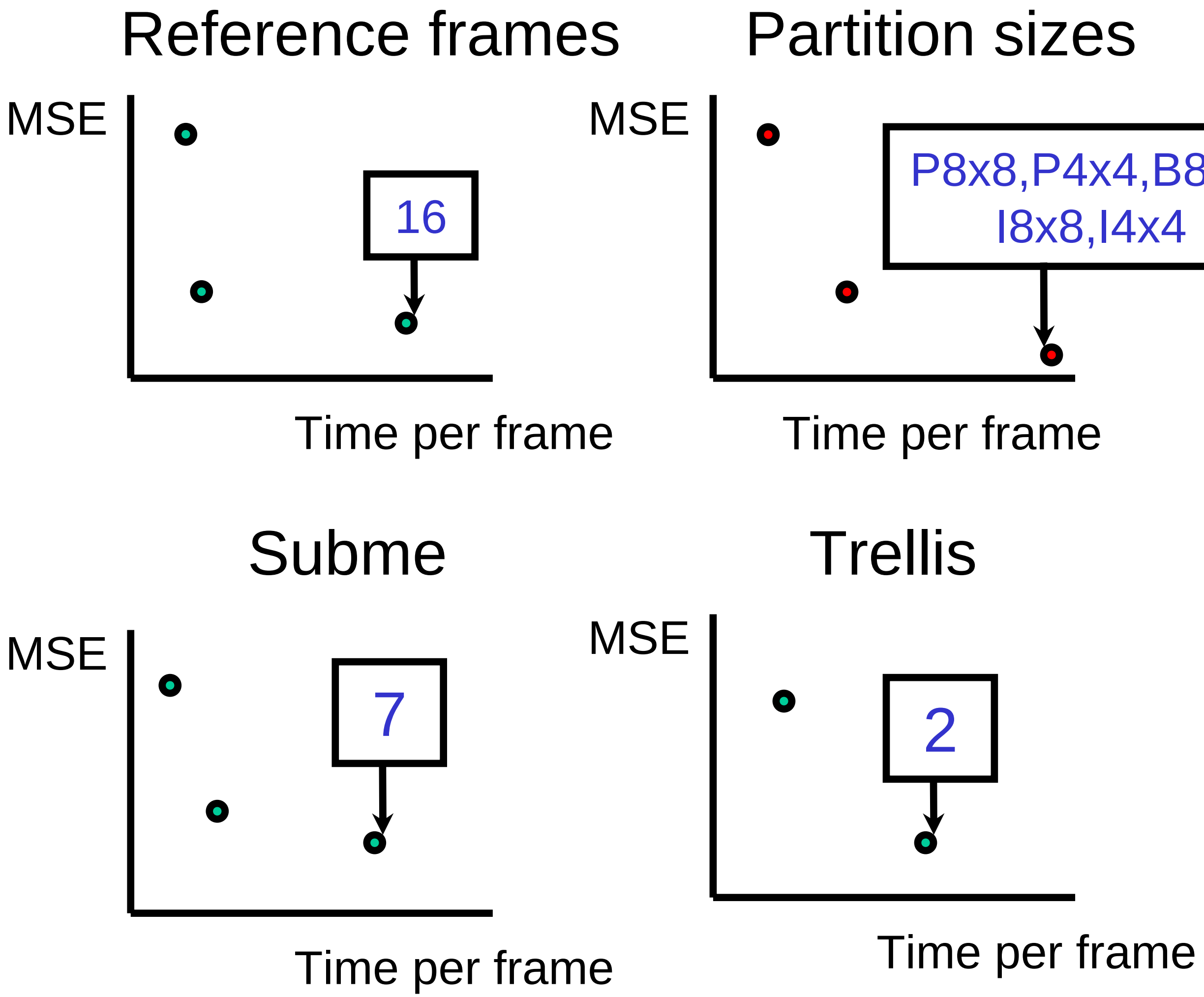


The process is repeated to obtain the final parameter setting

GBFOS-Iterative Algorithm

- More number of encodings
- Better performance
- Initialization and first iteration same as the GBFOS-Basic Algorithm

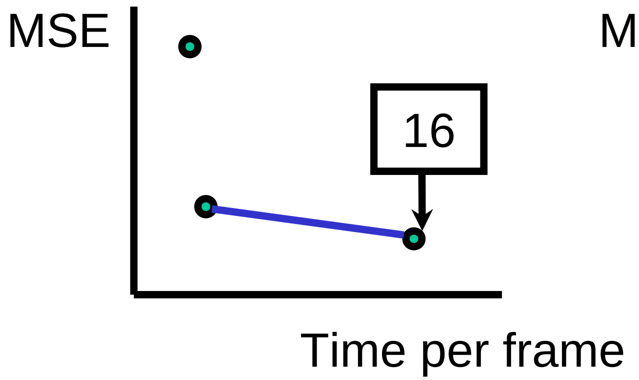
GBFOS-Iterative Algorithm



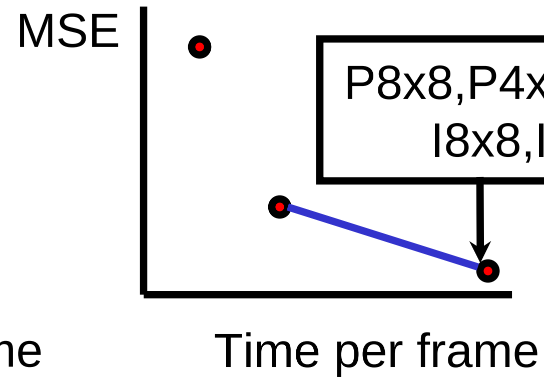
First parameter setting selected

GBFOS-Iterative Algorithm

Reference frames

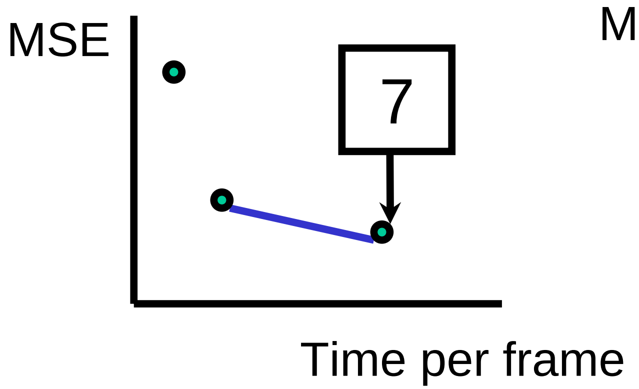


Partition sizes

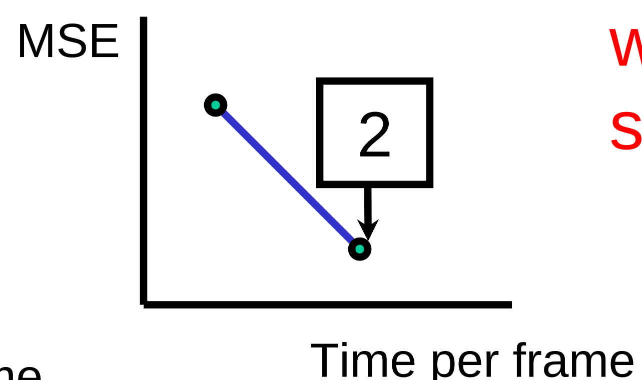


Compare slopes

Subme

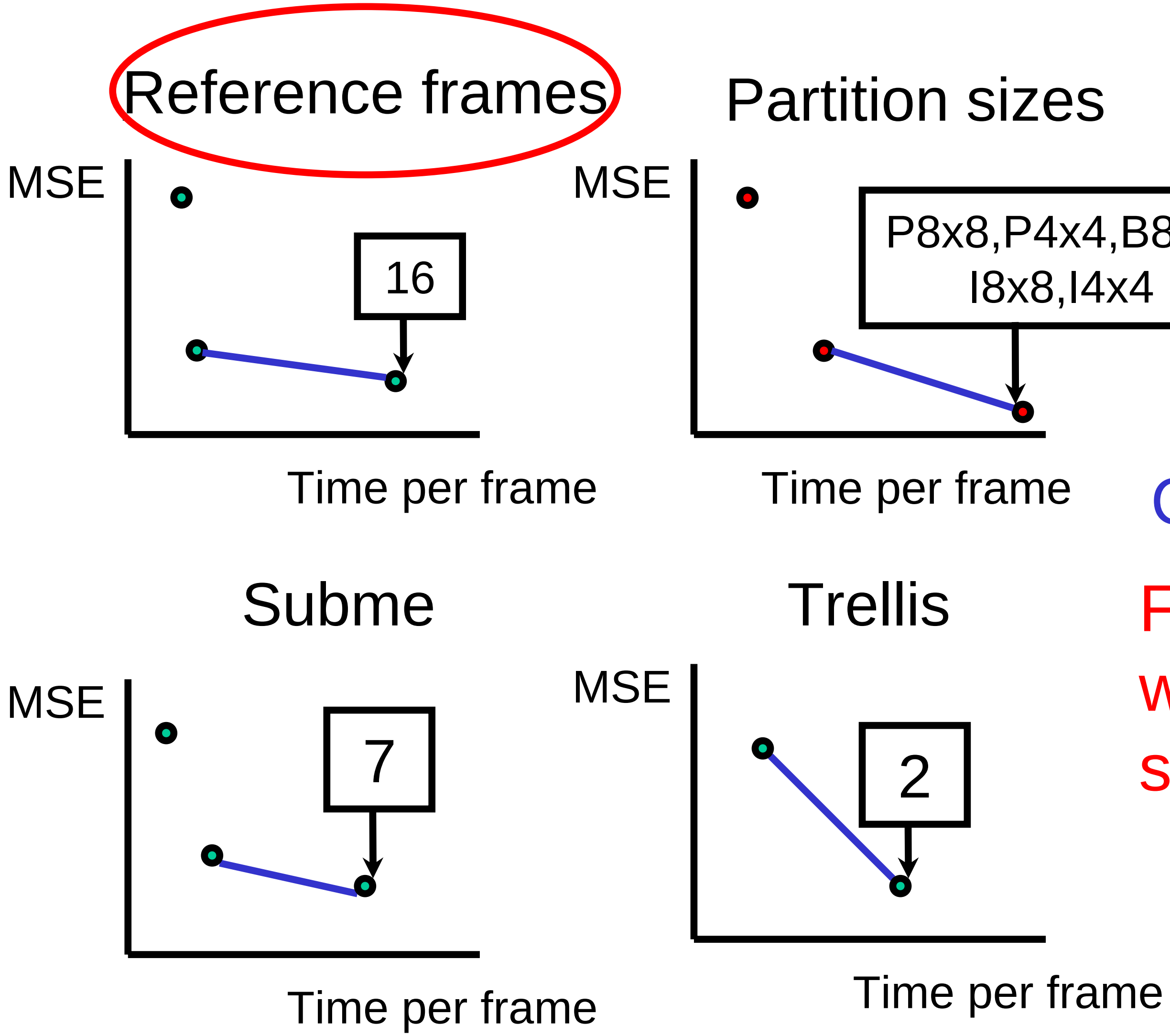


Trellis



Find the parameter with the least slope

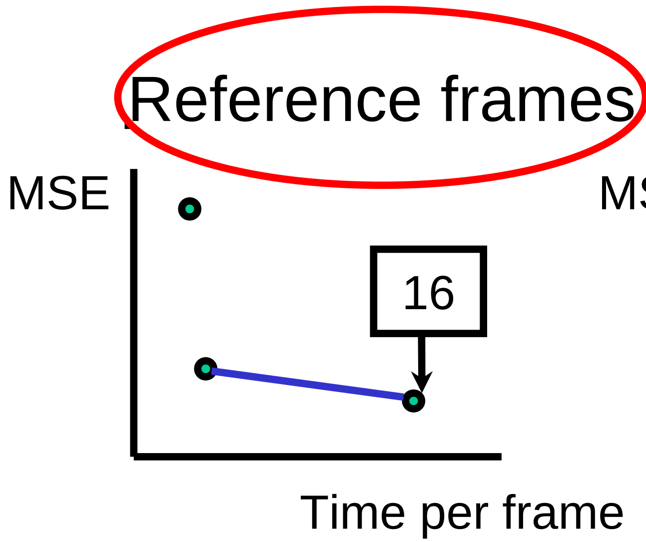
GBFOS-Iterative Algorithm

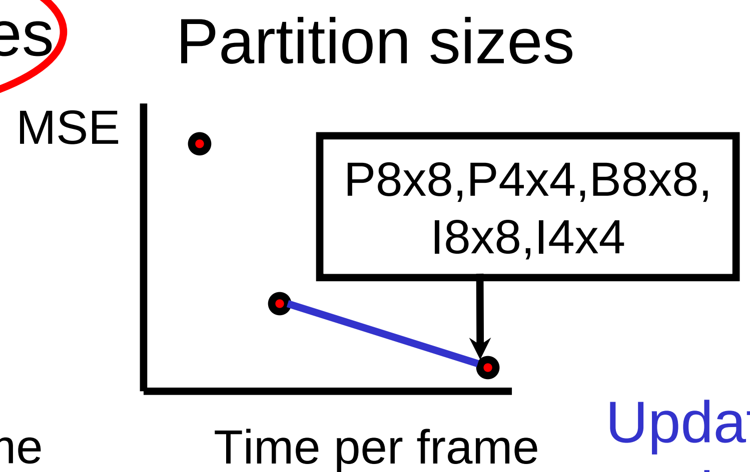


Compare slopes

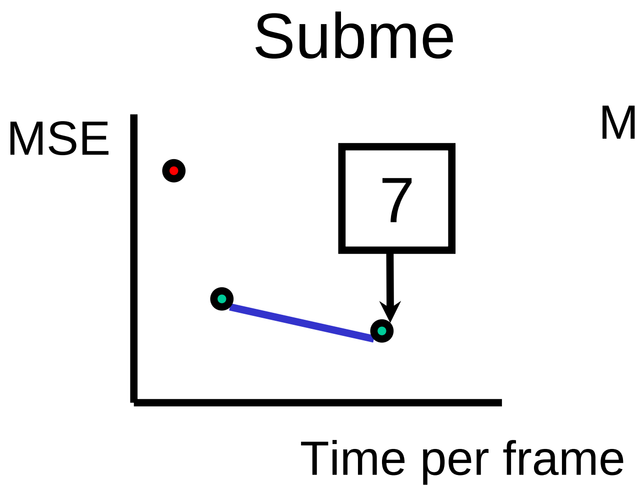
Find the parameter with the least slope

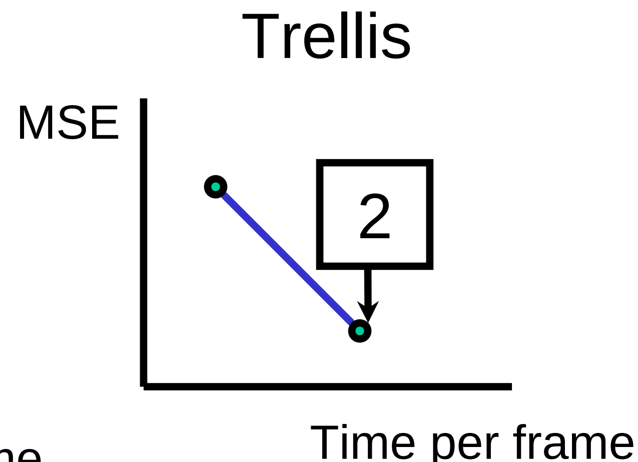
GBFOS-Iterative Algorithm





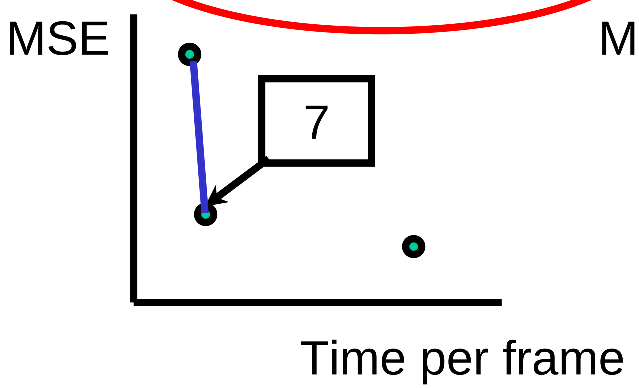
Update the slope and setting of Ref. frames



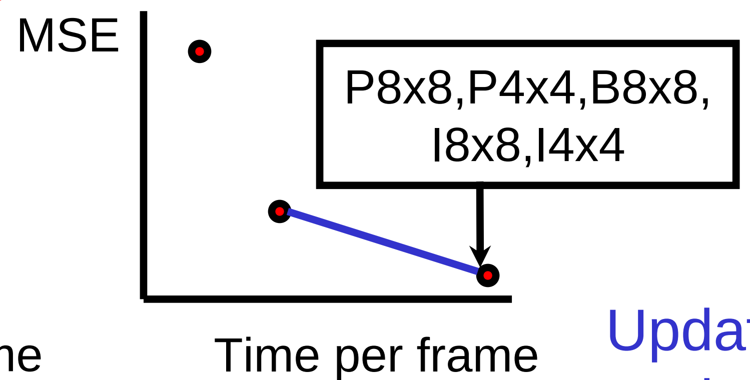


GBFOS-Iterative Algorithm

Reference frames

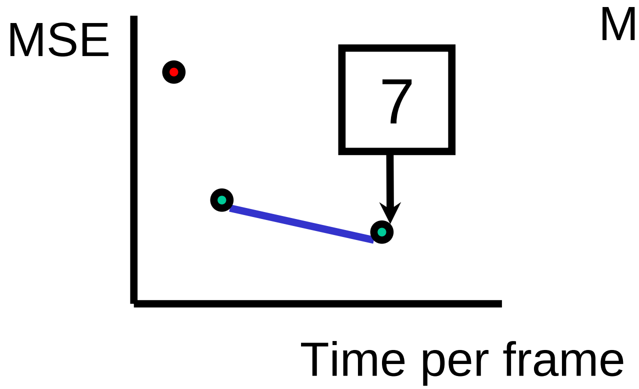


Partition sizes

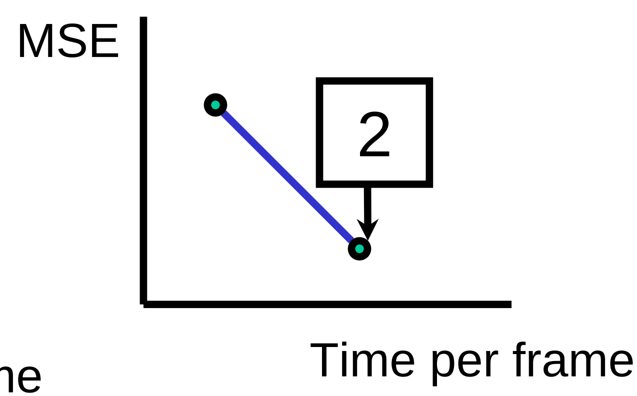


Update the slope and setting of Ref. frames

Subme

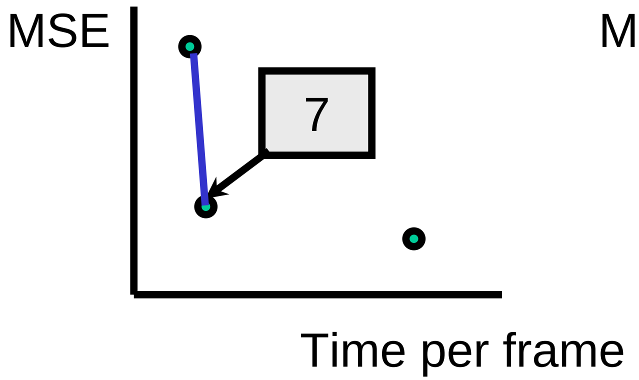


Trellis

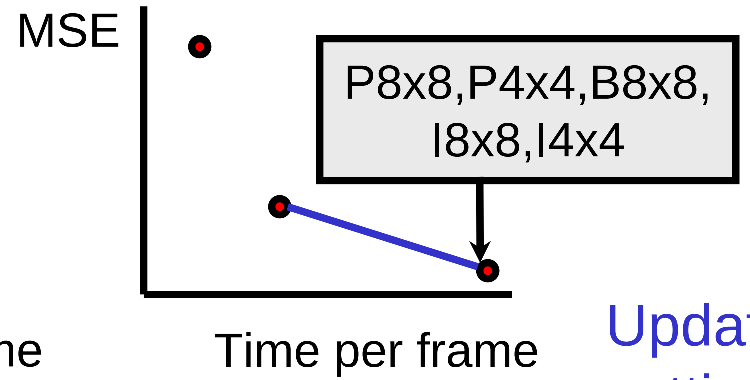


GBFOS-Iterative Algorithm

Reference frames

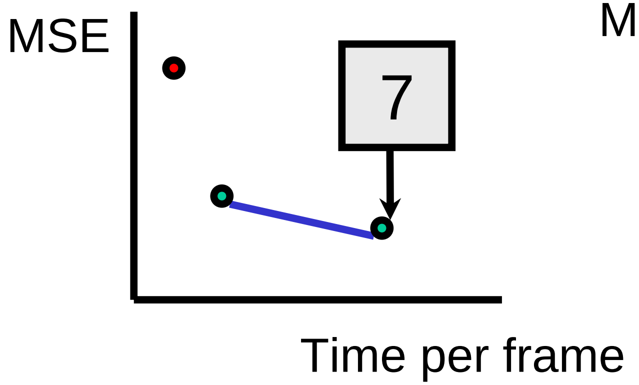


Partition sizes

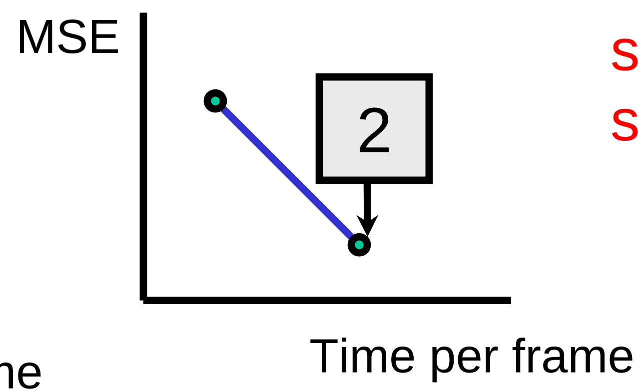


Update the slope and setting of Ref. frames

Subme



Trellis

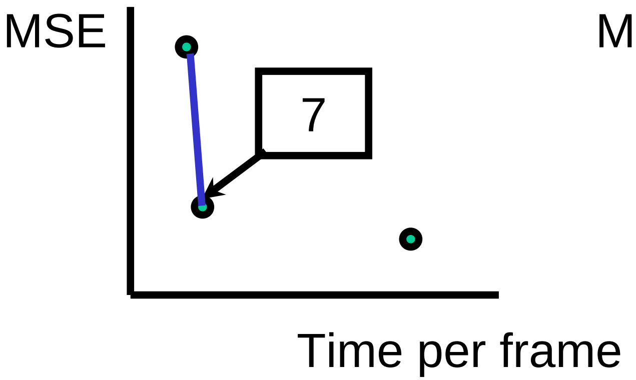


We now have the second parameter setting

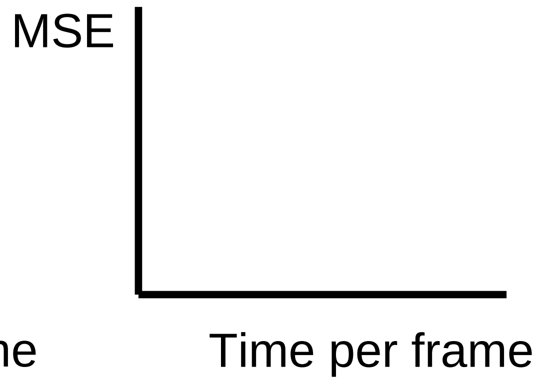
GBFOS-Iterative Algorithm

Second Iteration

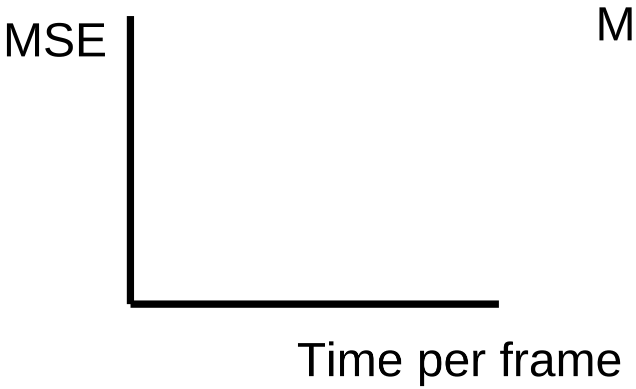
Reference frames



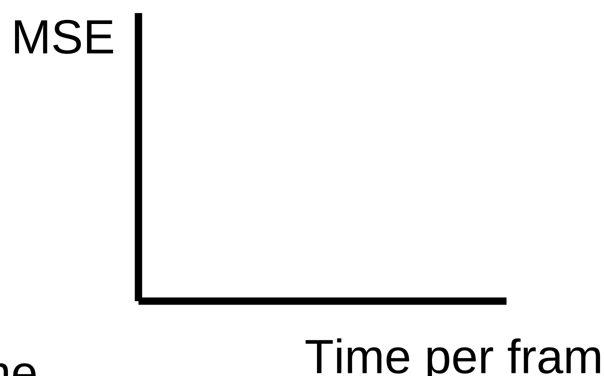
Partition sizes



Subme



Trellis



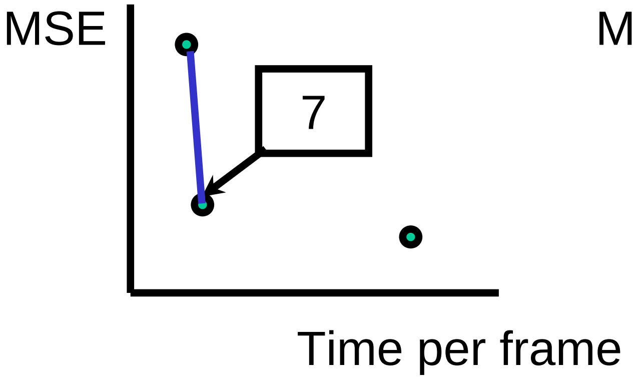
Regenerate the D-C plots for partition size, subme and trellis using Ref. frames = 7

Other parameters are obtained from the previous step

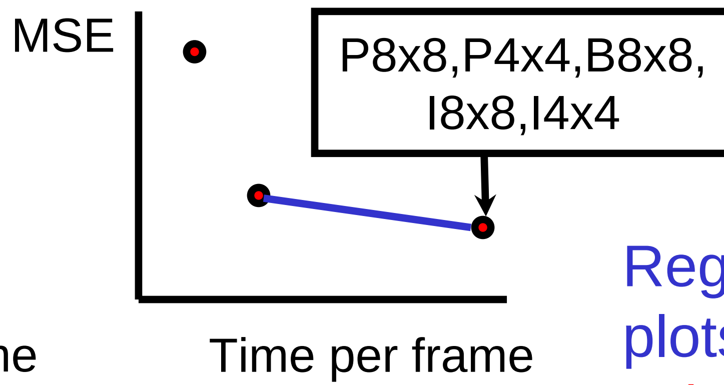
GBFOS-Iterative Algorithm

Second Iteration

Reference frames

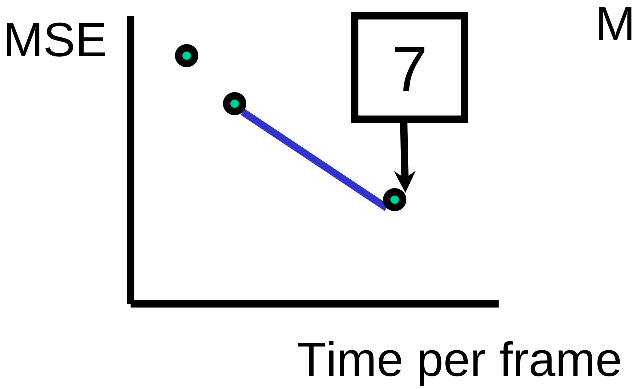


Partition sizes

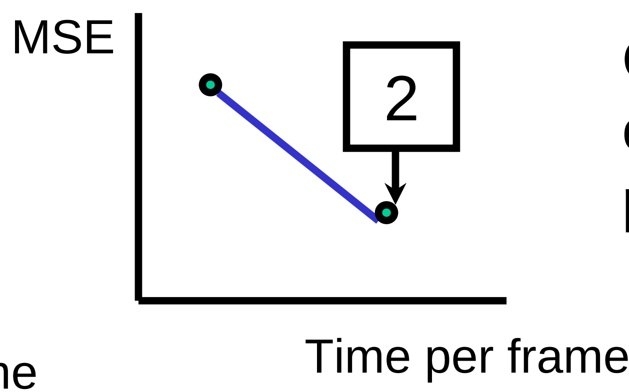


Regenerate the D-C plots for partition size, subme and trellis using Ref. frames = 7

Subme



Trellis

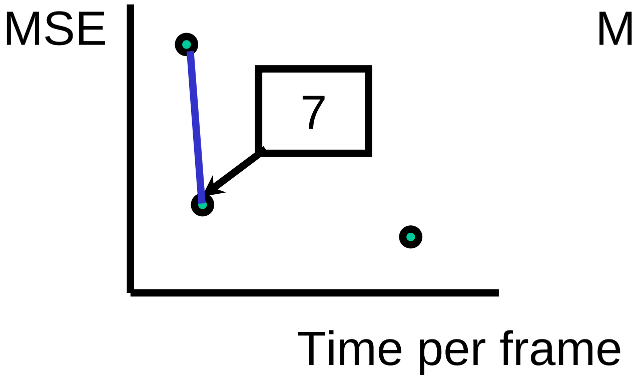


Other parameters are obtained from the previous step

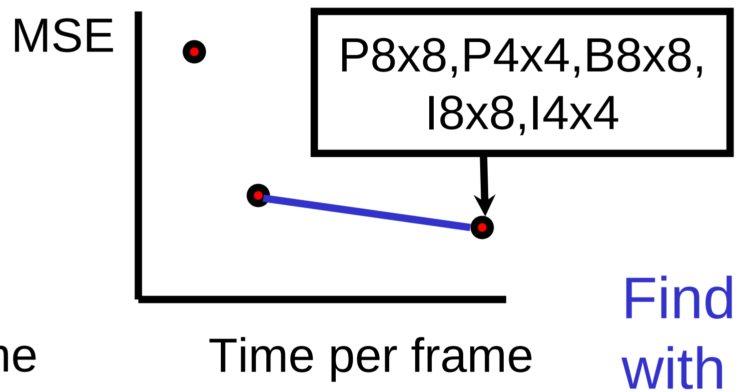
GBFOS-Iterative Algorithm

Second Iteration

Reference frames

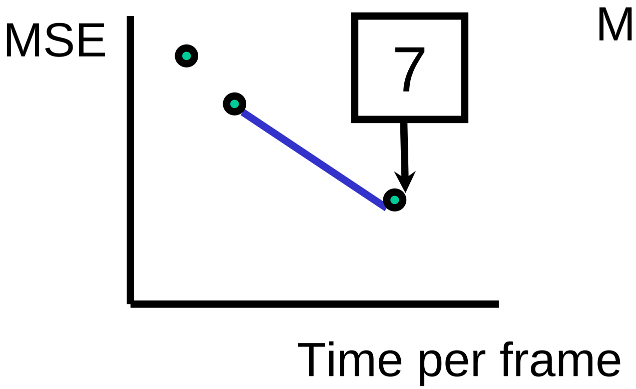


Partition sizes

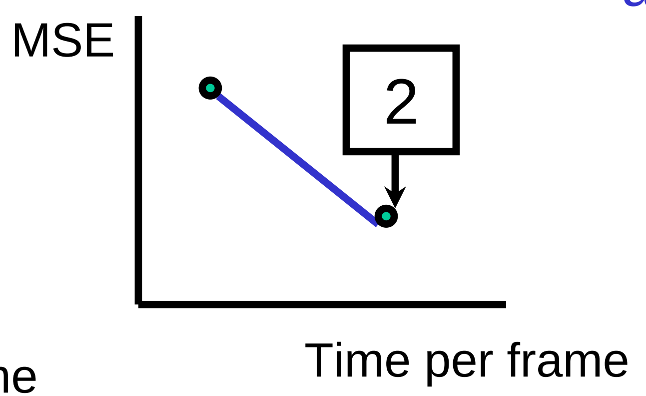


Find the parameter with the least slope and update its slope and setting

Subme



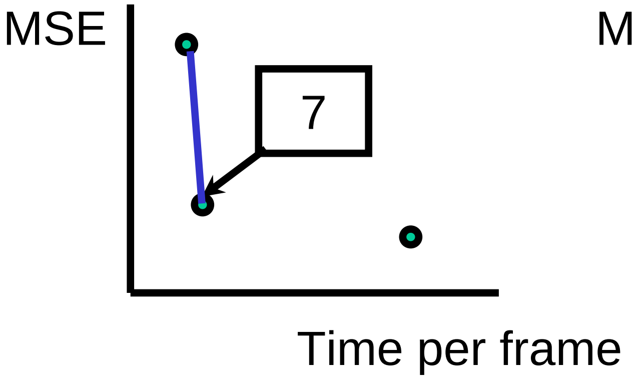
Trellis



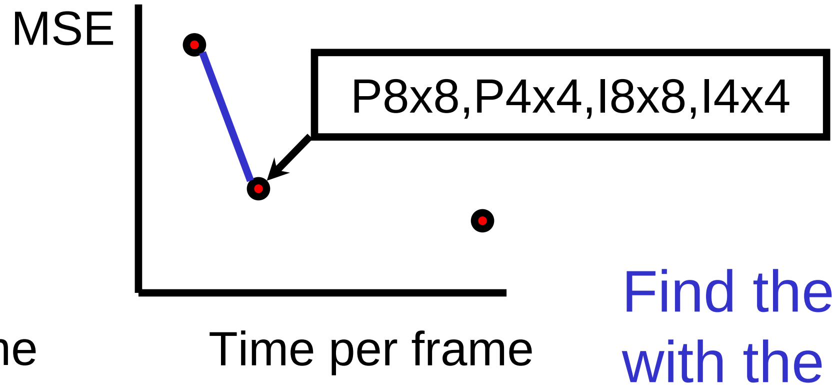
GBFOS-Iterative Algorithm

Second Iteration

Reference frames

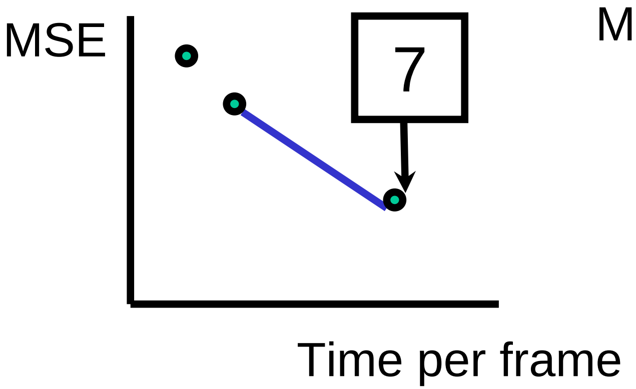


Partition sizes

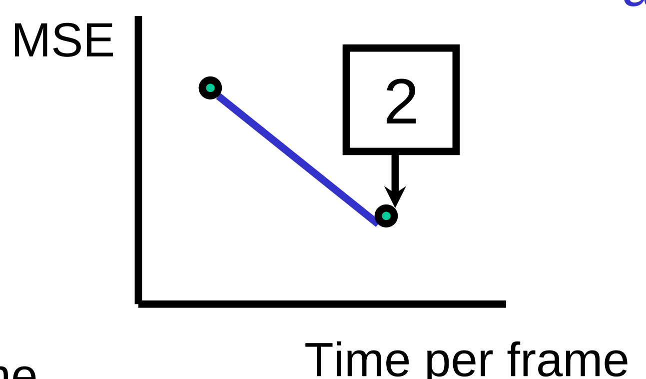


Find the parameter with the least slope and update its slope and setting

Subme



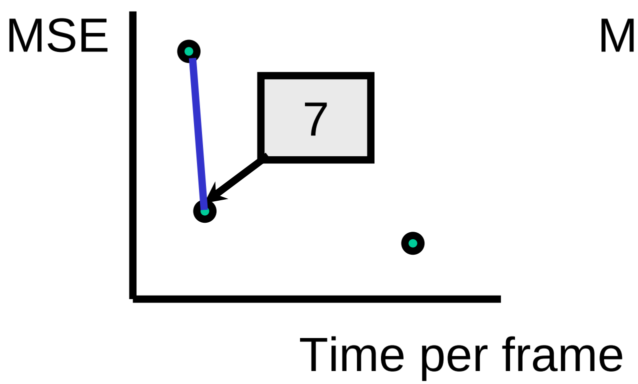
Trellis



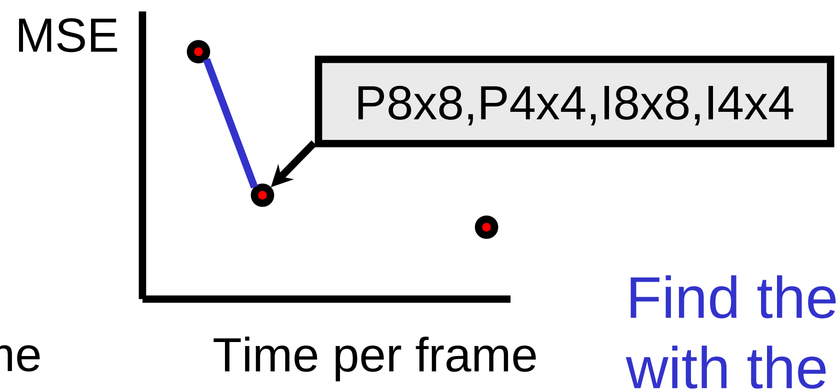
GBFOS-Iterative Algorithm

Second Iteration

Reference frames

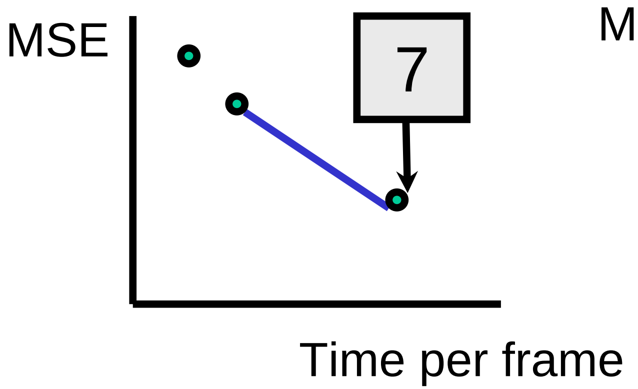


Partition sizes

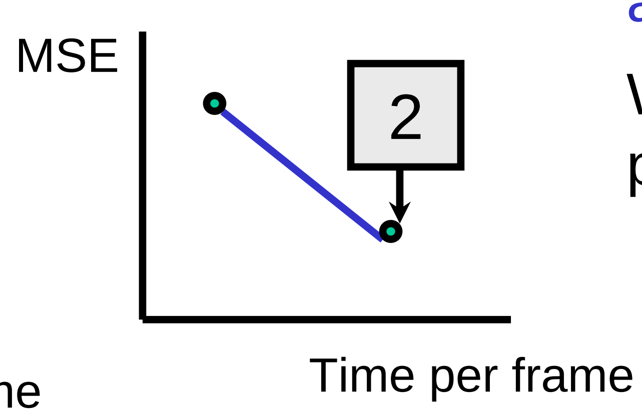


Find the parameter with the least slope and update its slope and setting

Subme



Trellis

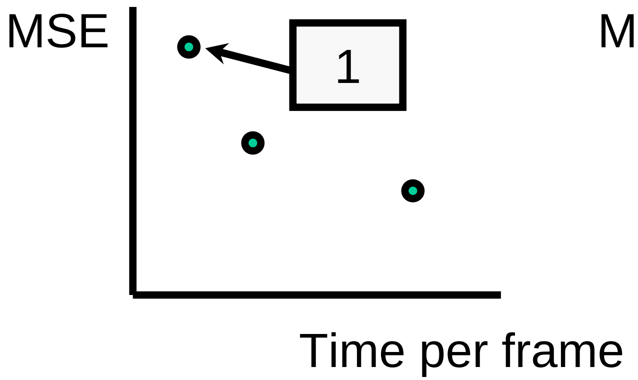


We now have the new parameter setting

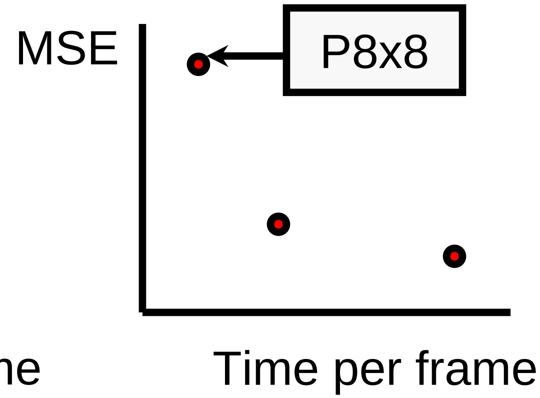
GBFOS-Iterative Algorithm

Final Iteration

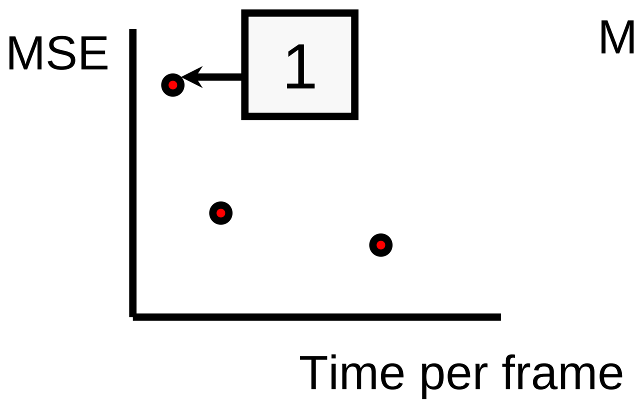
Reference frames



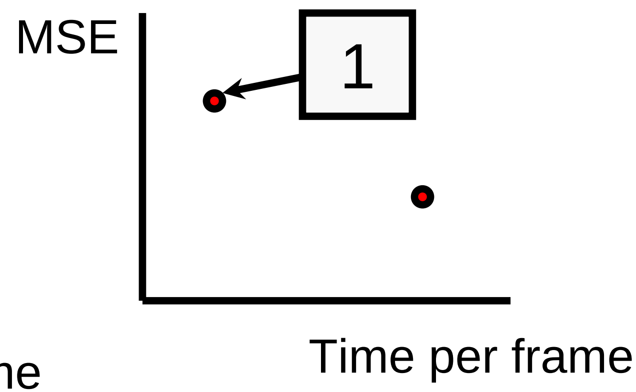
Partition sizes



Subme



Trellis



The process is repeated till we obtain the final parameter setting

Results

- In our test, we use 3 data sets – ASL-1 (10 QCIF files), ASL-2 (10 320x240 files) and Standard (15 QCIF files) and 3 bitrates – 30, 150 and 300 kb/s

- Four variable encoding parameters: number of reference frames (16 options), Partition sizes (10 options), Sub-pixel motion estimation (7 options) and trellis (3 options)

Results

- If $n_1,...,n_M$ are *M* parameter options, then the number of encodings per video sequence for

  Convex hull: $\prod_{i=1}^{M} n_i$ (product)

  GBFOS-basic: $1 - M + \sum_{i=1}^{M} n_i$ (sum!)

- In our example, the number of encodings for

  Convex hull = 3360 (constant)

  GBFOS - basic = 33 (constant) ≈ **1%** Convex hull

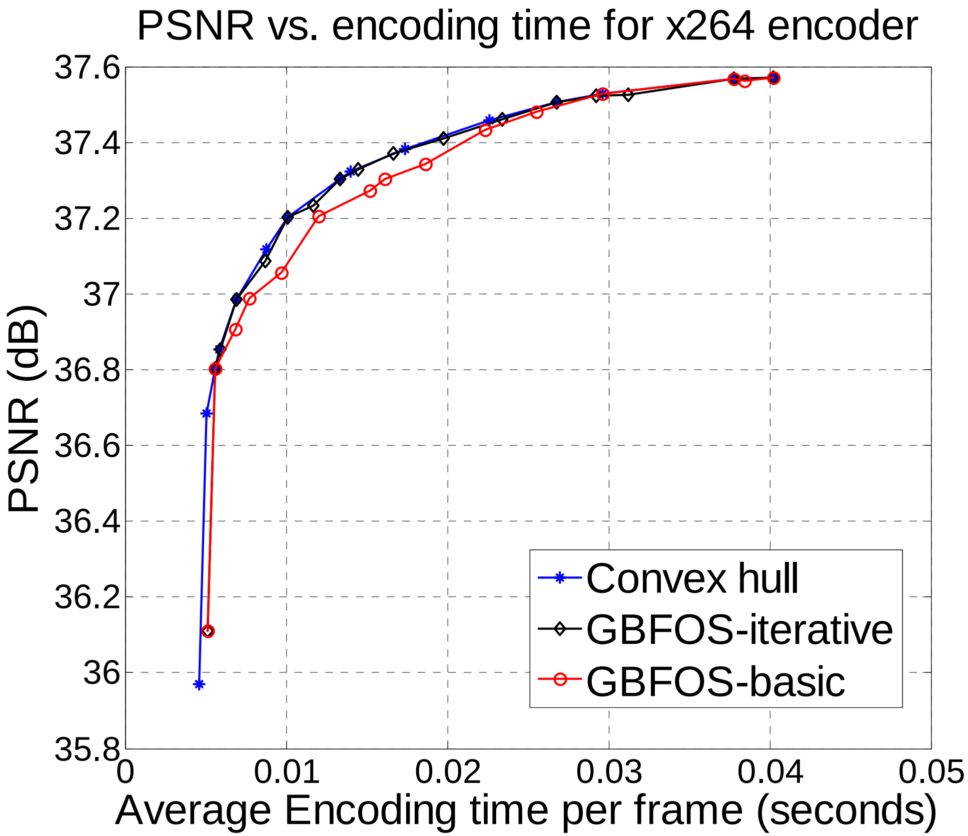
  GBFOS - iterative = 268 (maximum) ≈ **8%** of Convex hull

Results

- Maximum PSNR difference with respect to convex hull for:

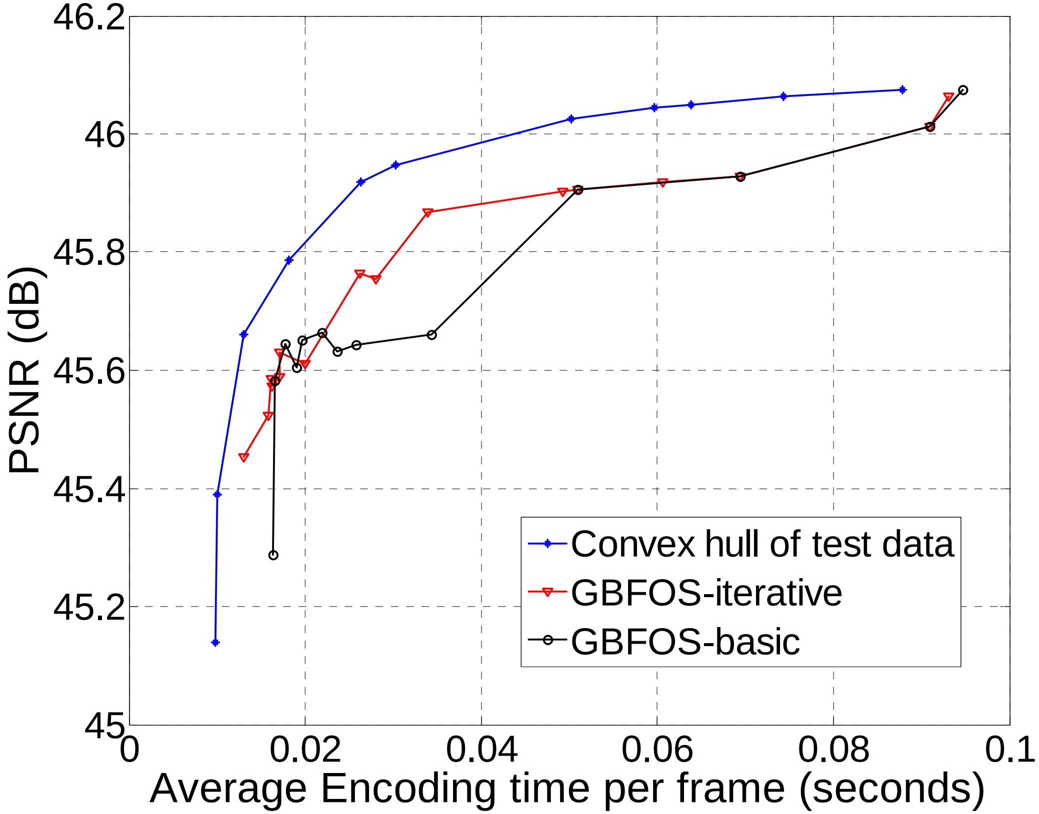
  GBFOS-basic = 0.707 dB

  GBFOS-iterative = 0.575 dB



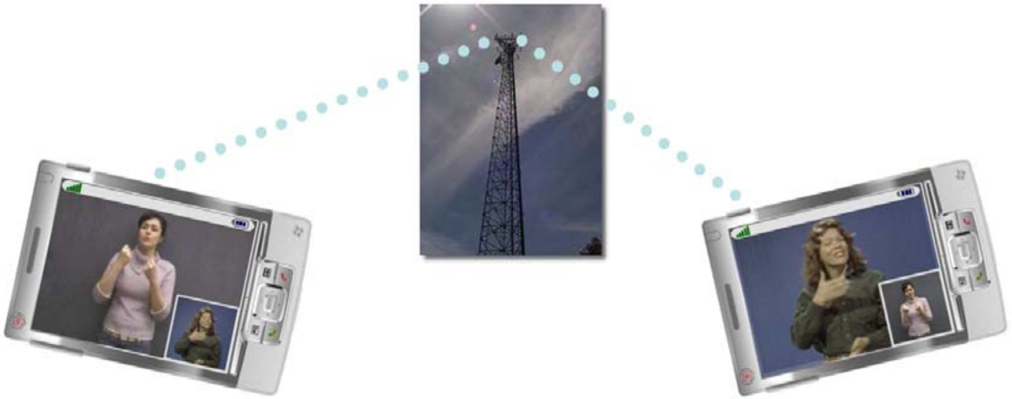
Results

- Robustness test done using 3 different (data set, training set) pairs at 3 different bitrates
- Max PSNR difference between either GBFOS algorithms and convex hull points **< 0.55 dB**



Application

- Video cell phones have limited processing speed
- Finding convex hull points is infeasible!
- GBFOS algorithms can be used for finding close-to-optimal encoder settings
- Goal: Real-time communication of American Sign Language videos over the GPRS network



http://www.cs.washington.edu/research/MobileASL/index.html

Conclusion

- Proposed two fast algorithms: GBFOS-basic and GBFOS-iterative for choosing encoder parameter settings
- GBFOS-basic and GBFOS-iterative algorithms take **1%** and **8%** of encodings required for obtaining the D-C convex hull points
- For same test and training data: Max PSNR difference for both algorithms **< 0.71 dB**
- For different test and training data: Max PSNR difference for both algorithms **< 0.55 dB**

Thanks!

- To National Science Foundation
- To Anna Cavender, Neva Cherniavsky and Loren Merritt