Efficient Block Based Motion Estimation

Deepak Turaga
Mohamed Alkanhal
Tsuhan Chen

Overview of Talk

- Motion Estimation
- Sub-Optimal Search Strategies
- Hierarchical Search Methods
- Spatial Correlation Based Searches

Motion Estimation

Conditional replenishment to remove temporal redundancy

Block Based Motion Estimation

- Break Frame into blocks, find best match for block in previous frame. Send only motion vectors and residue

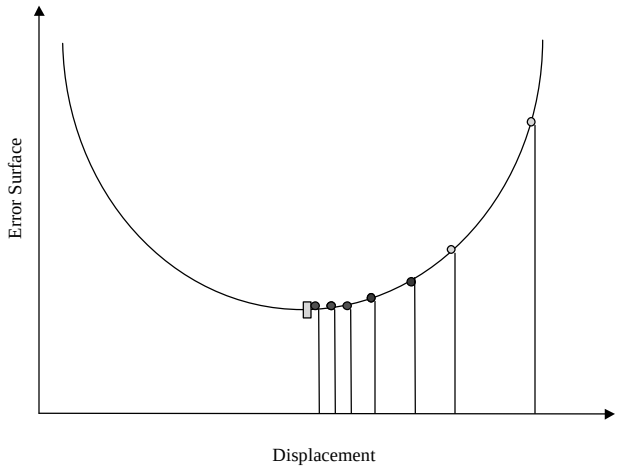
Block Based Motion Estimation Algorithms

- Full Search : Optimal, but computationally expensive
- Sub-Optimal Search Strategies
  - Sacrifice quality of motion compensation for speed.
  - Three Step Search, Four Step Search, Cross Search, Orthogonal Search, Hierarchical Search etc.

Search Strategies - Type I

Pick a large step size around the starting point.

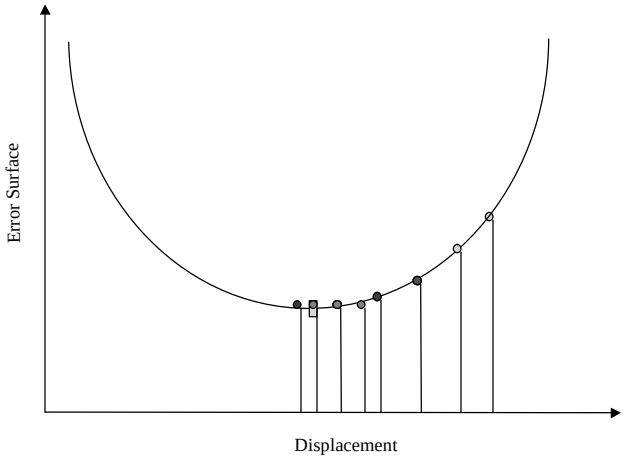
At every stage reduce the step size and move in the direction of the best match.



Search Strategies - Type II

Move in direction of best match.

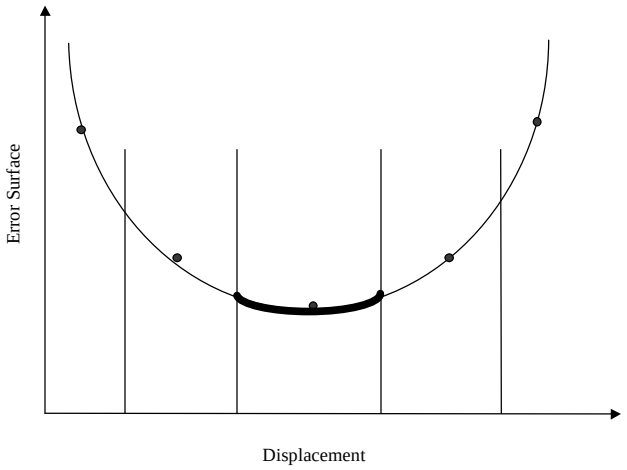
Reduce step size only on overshoot.



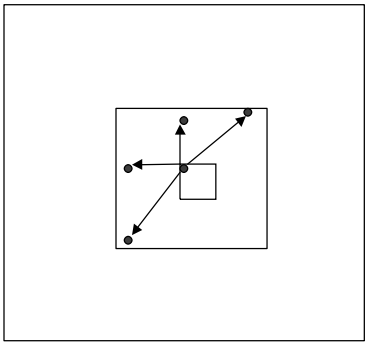
Search Strategies - Type III

Divide search space into many regions. Pick a center point for each region.

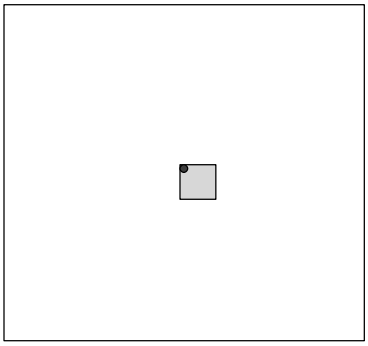
Do a full search over region with best center.



Best Matches and Motion Vectors



Previous Frame

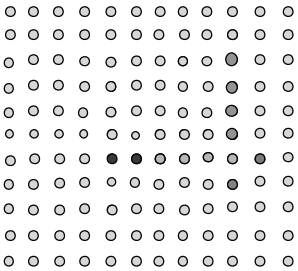


Current Frame

Sub-Optimal Search Strategies

One at a Time Search

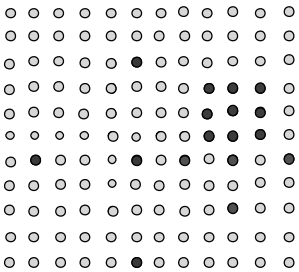
Look horizontally first, then look vertically. Very fast, but not very good quality.



Sample Convergence Path

2-D Log Search

Use '+' pattern. Reduce step size at every stage. Move center to best match

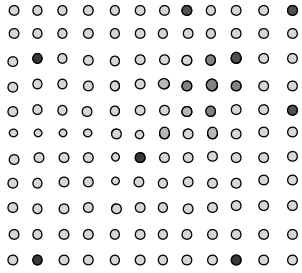


Sample Convergence Path

Sub-Optimal Search Strategies

Cross Search

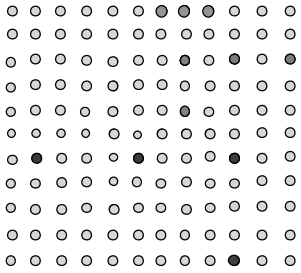
Similar to 2-D Log Search. Use a cross pattern instead.



Sample Convergence Path

Orthogonal Search

Alternate between horizontal and vertical directions. Move center to best match.

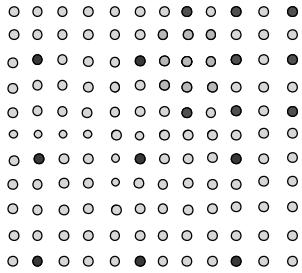


Sample Convergence Path

Sub-Optimal Search Strategies

Three Step Search(TSS)

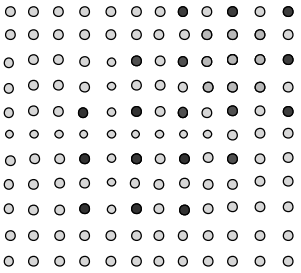
Do a coarse to fine search. Move in the direction of best match. Good when best match far from center of search.



Sample Convergence Path

Four Step Search(FSS)

Start with a fine step. Move in the direction of best match. Good when best match close to center of search

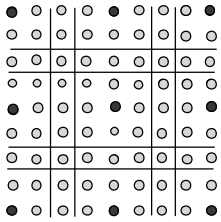


Sample Convergence Path

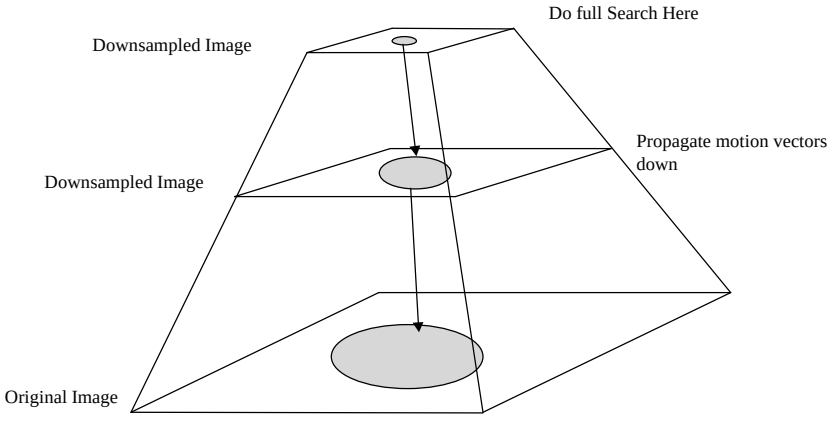
Sub-Optimal Search Strategies

Binary Search

Partition the search space into smaller regions and pick one region based on which of the initial blocks selected has the smallest MAD.

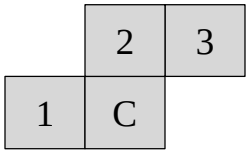


Hierarchical Search Strategies



Spatial Correlation Based Searches

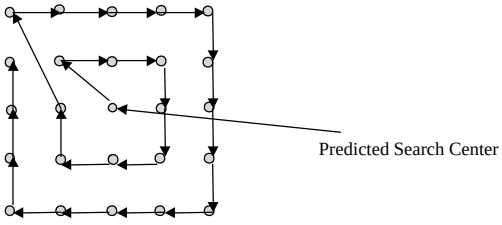
- Exploit spatial correlation within frames to narrow down the search.
- Blocks belonging to one object tend to move together.
- Use information from neighboring blocks to predict location of best match for current block.



- Use 1,2 and 3 as predictors for block C.

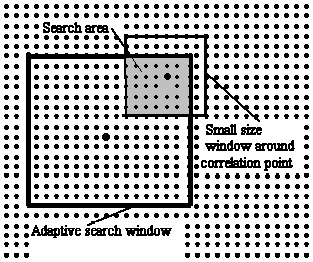
MAD Based Spiral Search

- Predict motion vector and MAD threshold for current block using predictor blocks.
- Move center of search to predicted motion.
- Do a spiral search around new center till MAD smaller than predicted threshold.
- Have to update to avoid accumulation of error.



Adaptive Window Size Search

- Use predictors to predict center of search and size of window around search center.
- Do a search over the intersection of original search window and new search window.
- Need to do regular update.



Majority Voting Schemes

- Choose between different algorithms adaptively. Combine their best features.
- Use spatial correlation information to choose algorithm.
- e.g. FSS better for small motion while TSS better for large motion. If predictors have large motion vectors pick TSS, else pick FSS.

Spatial Correlation Based Searches

- Reduce Search Space by moving search center to predicted best match.
- Do an efficient search over the reduced search space through choice of algorithm.
- Save in bits for motion vectors (MV for H.263 are differentially coded)
- Do well in terms of the “speed-quality-bitrate” tradeoff