Adaptive Motion Compensation Without Blocking Artifacts

Timothy B. Terriberry

The Problem

- Suppose we have a transform stage with no blocking artifacts (wavelets, lapped transforms...)
- Want to avoid blocking in motion compensation
- Overlapped Block Motion Compensation (OBMC) works great, but...
  - Non-overlapped Block Match Algorithms (BMA) get a lot of gains from supporting variable block sizes
  - Not obvious how to handle variable sizes with OBMC

Previous Work

- "Usual" approach is to overlap with the smallest block size (Dirac, Qualcomm's recent T13-SG16-C0806 submission, etc.)
  - Not really "variable size"
  - Just better entropy coding for "fixed size"
- Zhang et al. (1998) did "adaptive repartitioning"
  - Split large blocks until overlap compatible with their smaller neighbors
  - Might still use small windows with large blocks
  - How do you search?

Adaptive Repartitioning

- Still, results were encouraging...

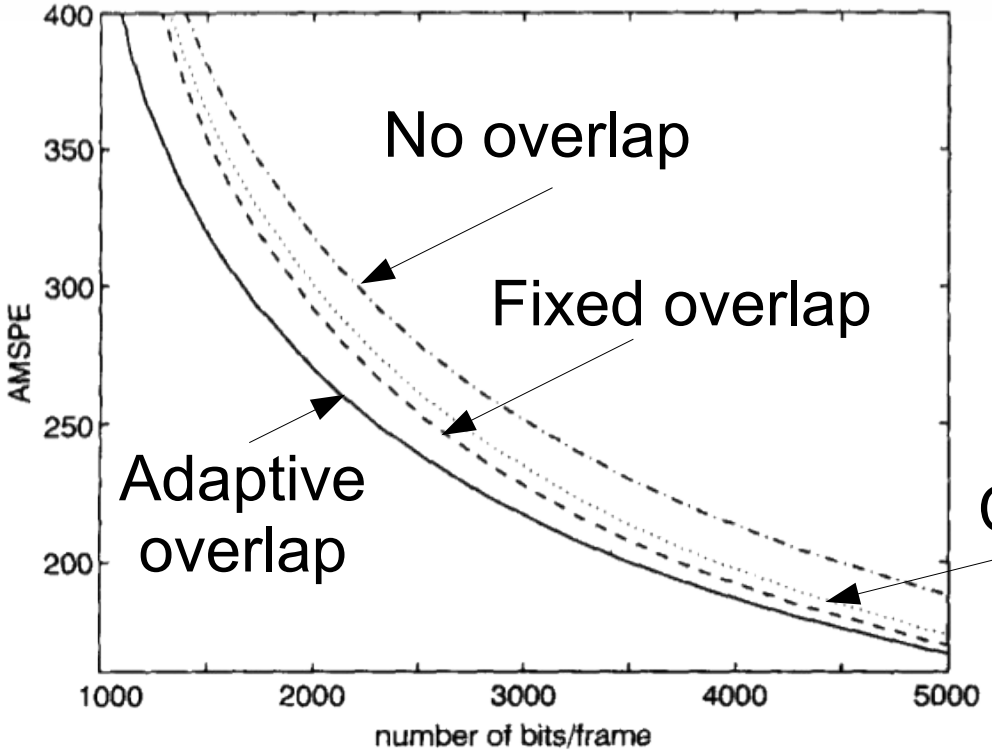


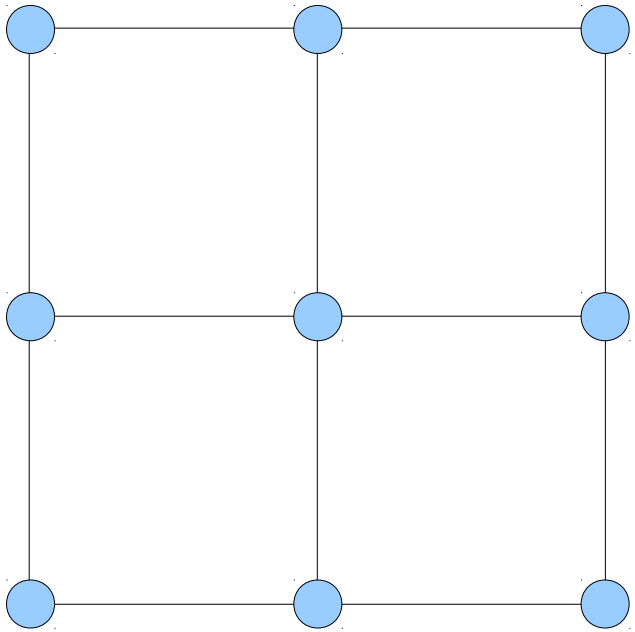
Fig. 8 *Rate-distortion performance for football sequence (VSBMC)*
VSBMC
VSBMC + GOBMC
VSBMC + VRP
VSBMC + AVRP

Proposed Approach

- Restrict adjacent block size to differ by no more than a factor of 2
- Use a 4-8 mesh to impose the restriction
  - Data structure used for surface simplification in computer graphics
  - Has a good search algorithm
- Define interpolation rules based on this restriction

4-8 Mesh

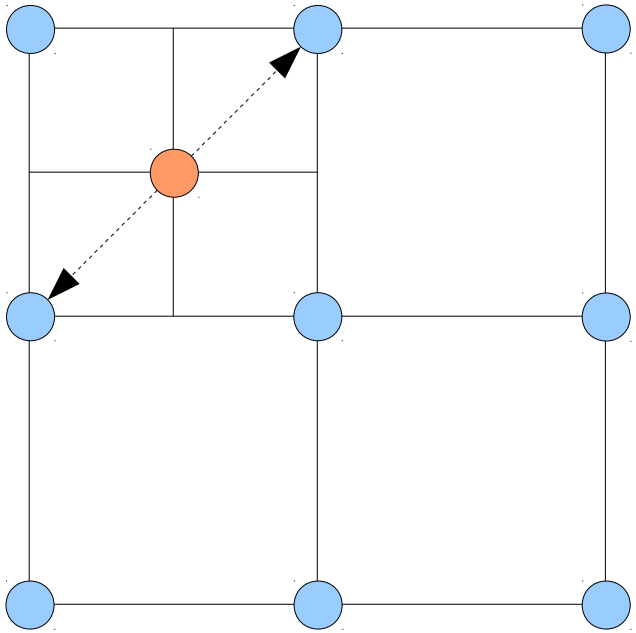
- Start with the largest block size



Level 0

4-8 Mesh

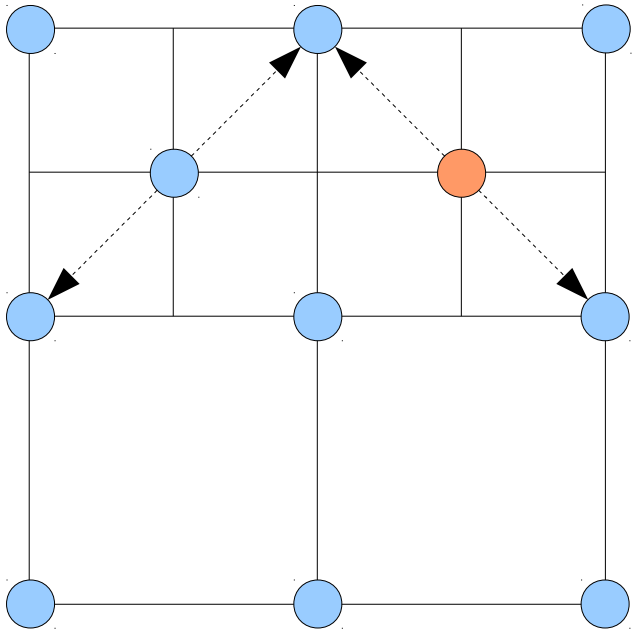
- Odd levels: can add new MVs to block centers
  - Only if *two parents* are in the grid (always at L1)



Level 1

4-8 Mesh

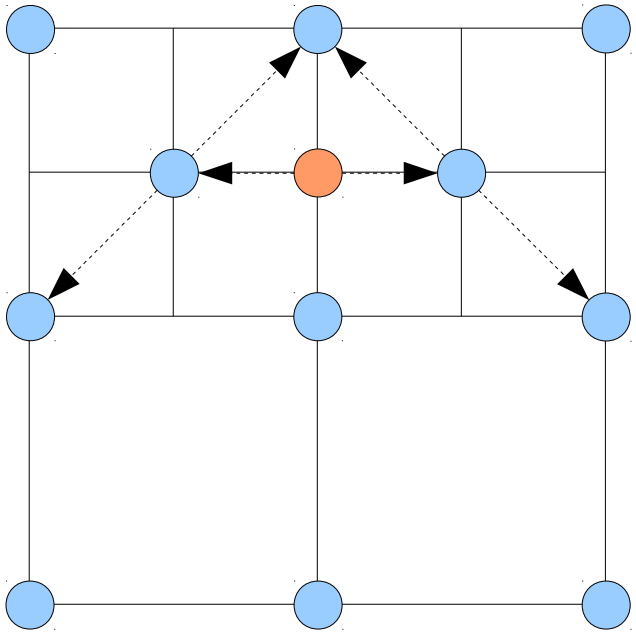
- Odd levels: can add new MVs to block centers
  - Only if *two parents* are in the grid (always at L1)



Level 1

4-8 Mesh

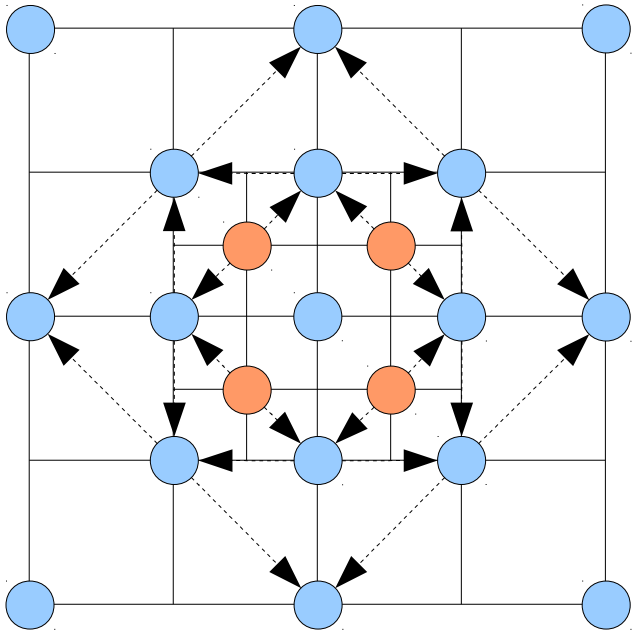
- Even levels: can add new MVs to block edges
  - Again, only if *two parents* are in the grid



Level 2

4-8 Mesh

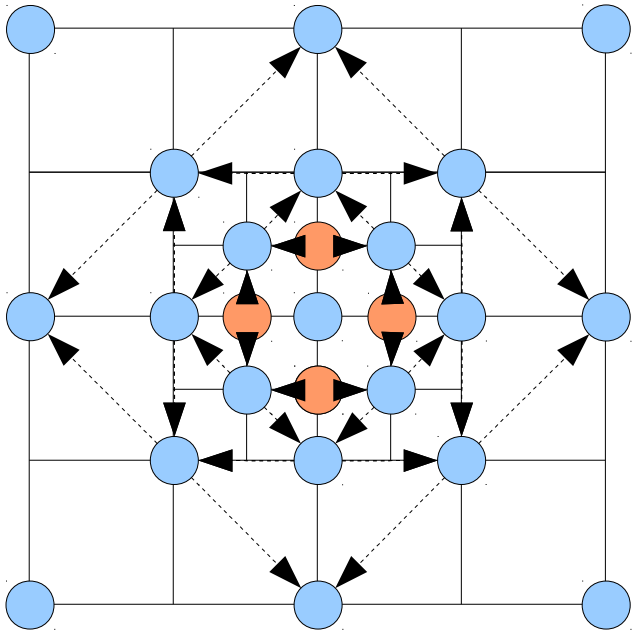
- And so on...



Level 3

4-8 Mesh

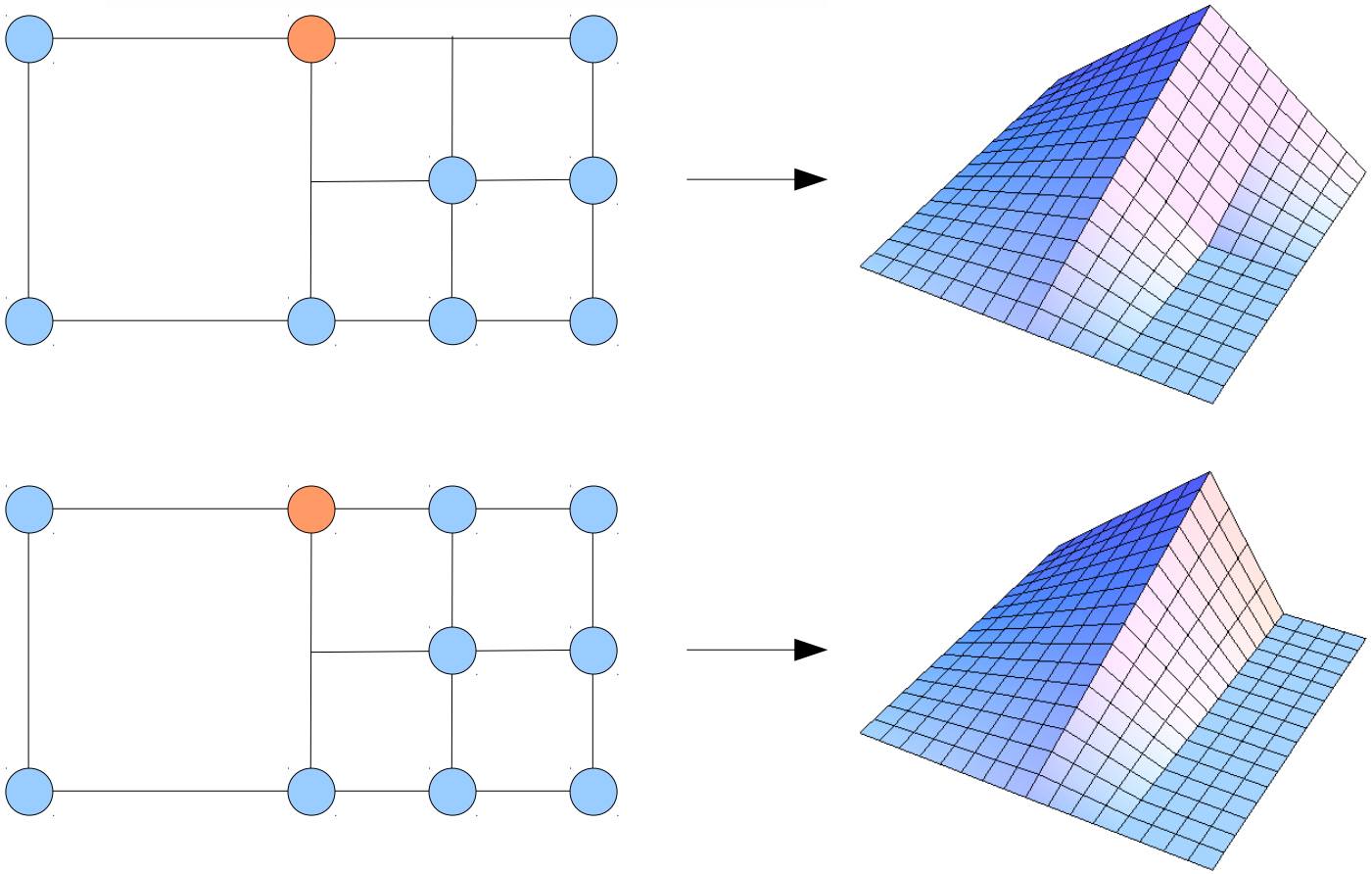
- And so on... and so on...



Level 4

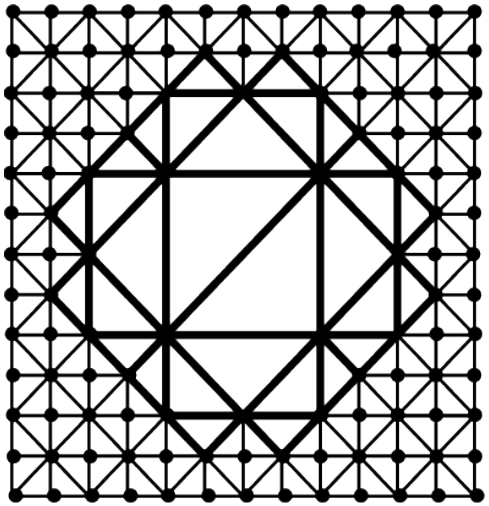
Blending Windows for Mismatched Block Sizes

- Simple sums of bilinear weights

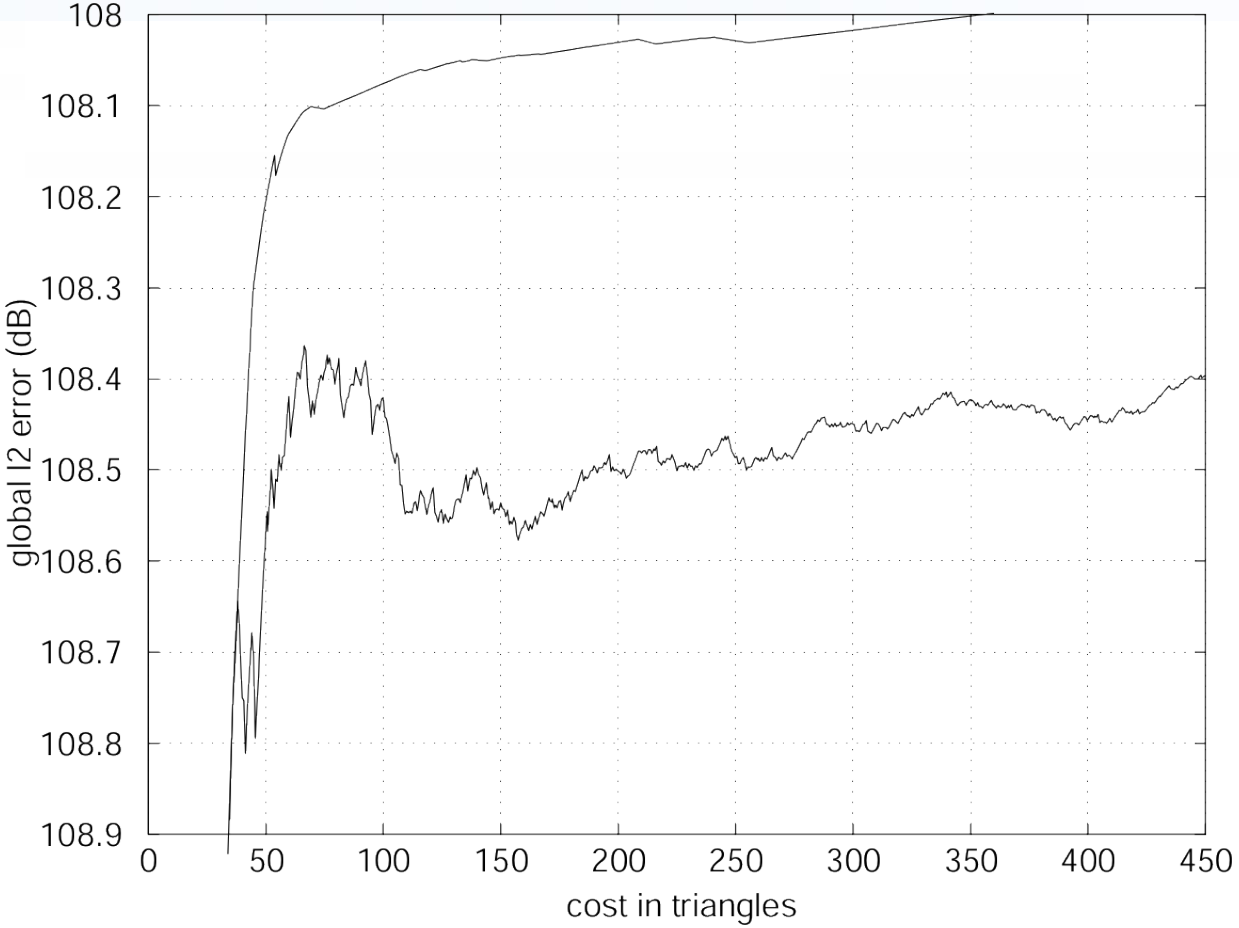


Block Size Decision

- Greedily inserting/removing leaves does poorly
  - Objective not monotonic
  - May have to add several MVs that regress R/D before you can add one that helps a lot
- “General Decimation” (Balmelli 2001)
  - Bottom-up, but...
  - Try deleting *every* MV in the grid
    - If it has children, delete them, too

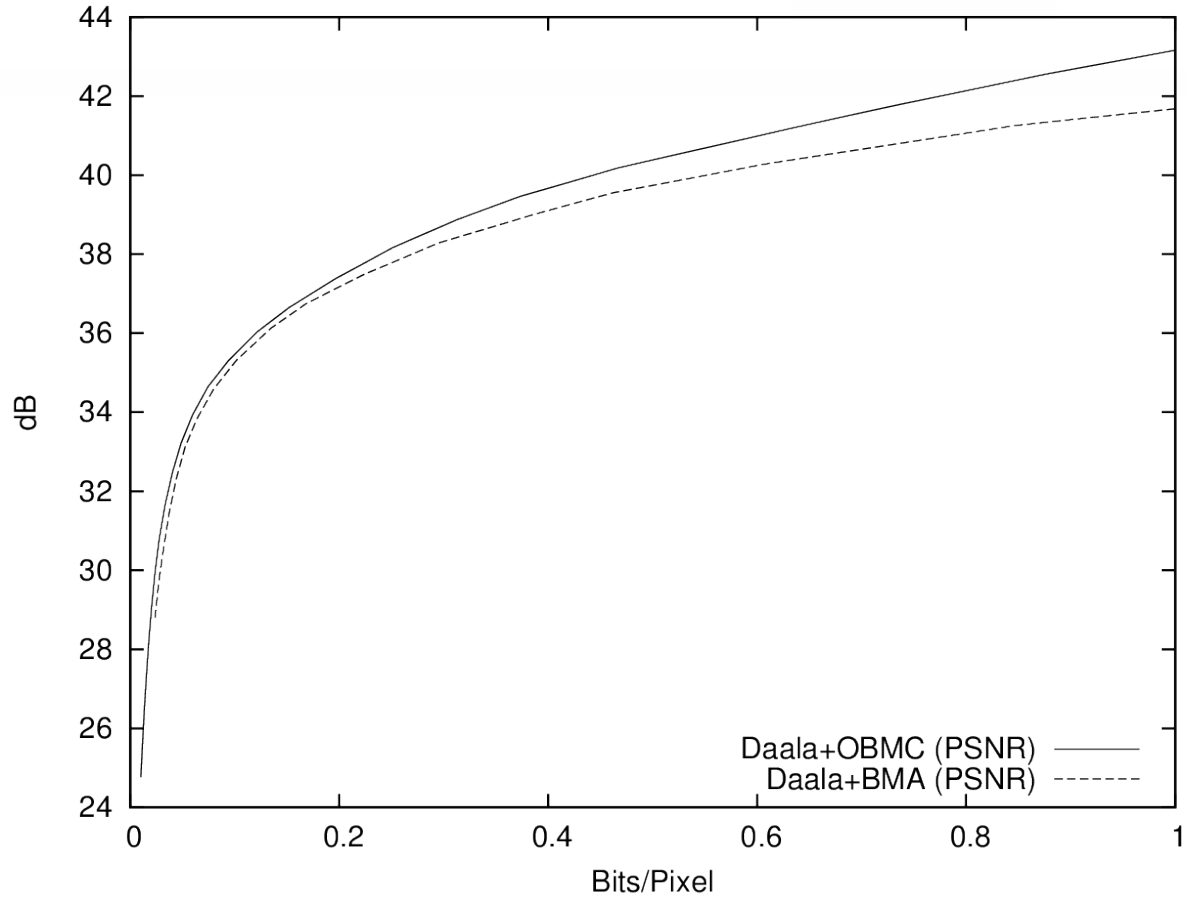


General Decimation



Greedy vs. General Decimation applied to surface simplification (Balmelli 2001)

New Scheme vs. Variable Block-Size BMA

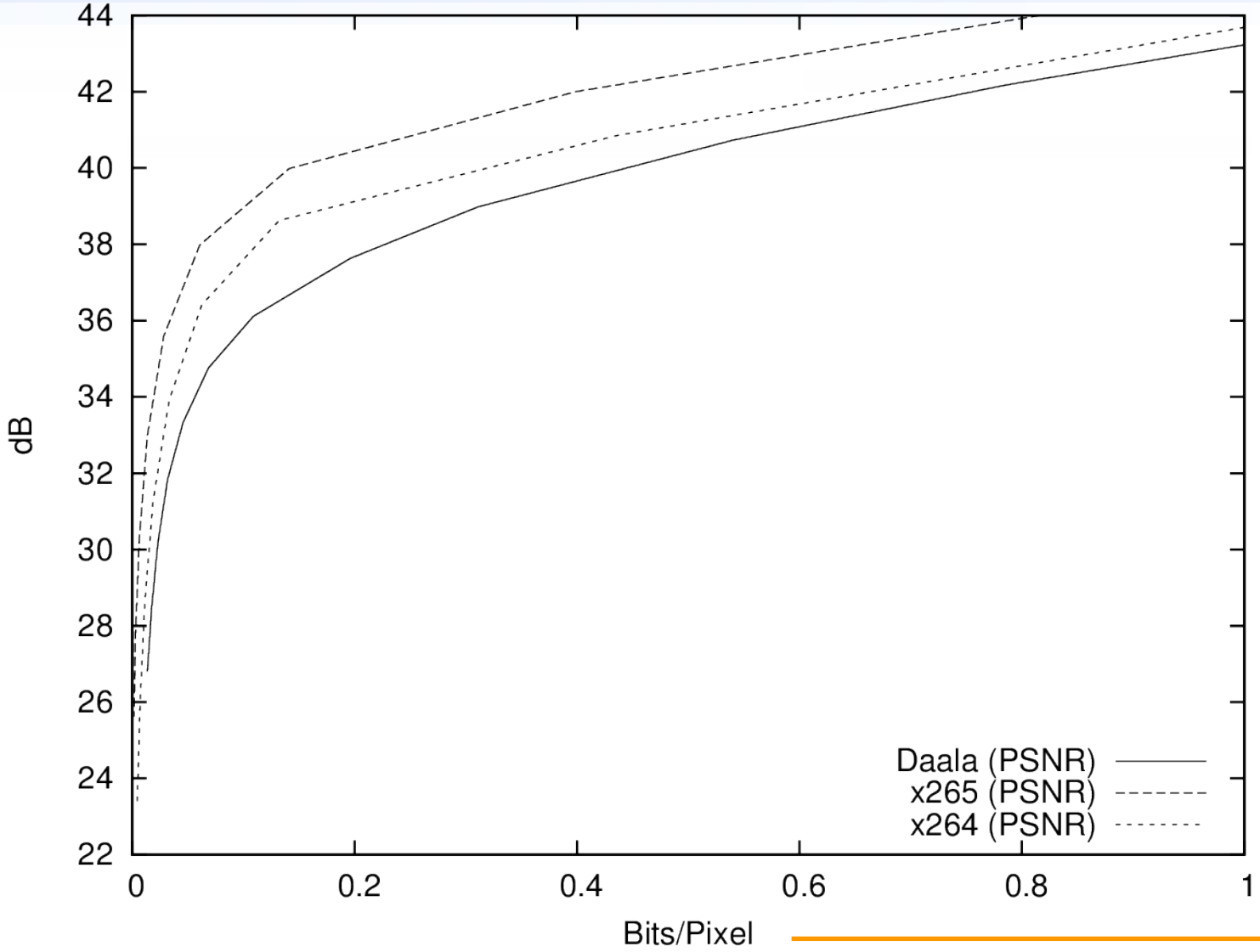


(Quick 2-hour hack experiment, not in paper)

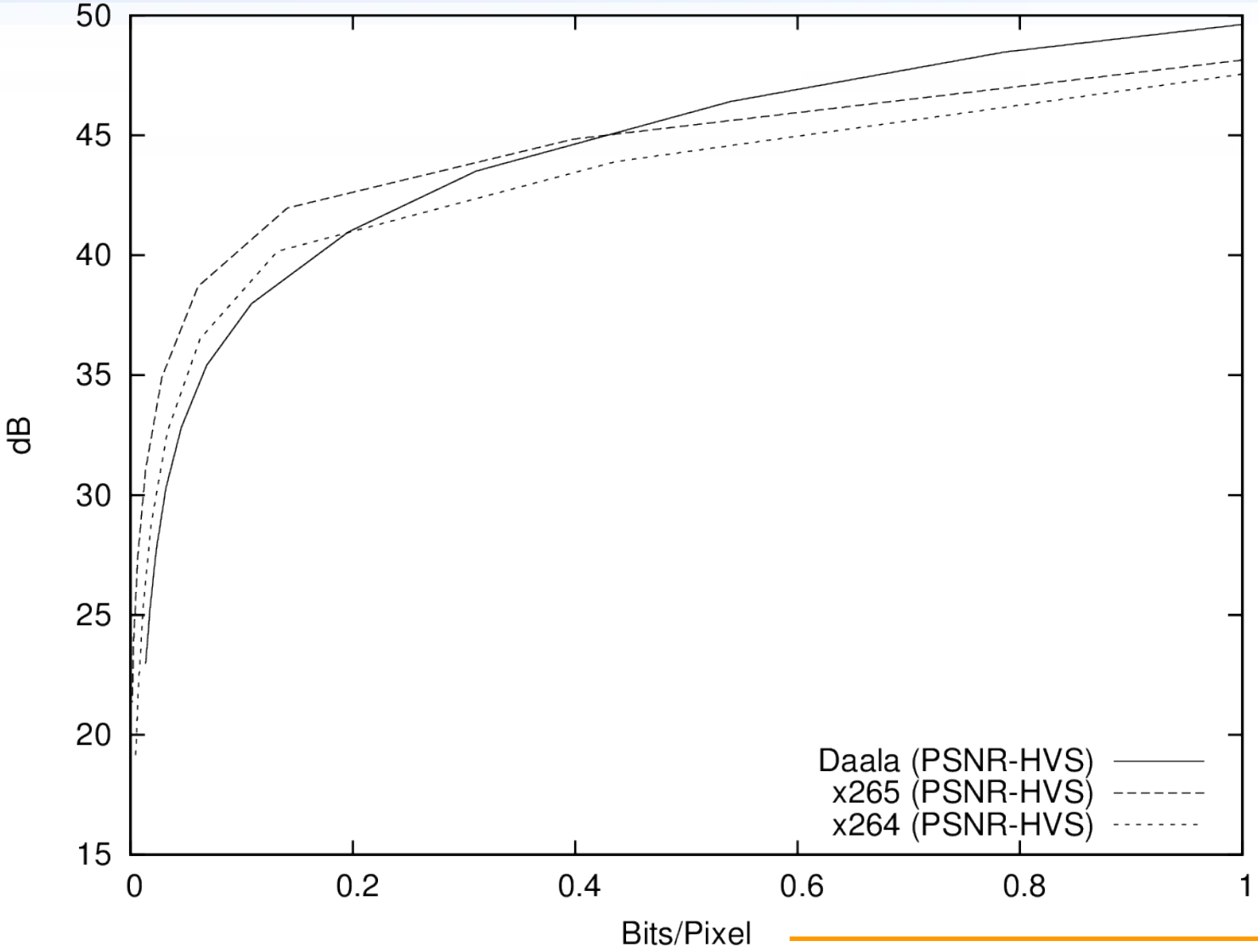
Future Work

- Can remove restriction on adjacent block sizes
  - Still need to keep edges between different sizes “unsplit”
  - No longer a 4-8 mesh
- Larger block sizes (currently only support 16x16)
- Better MV prediction
- Better encoder search

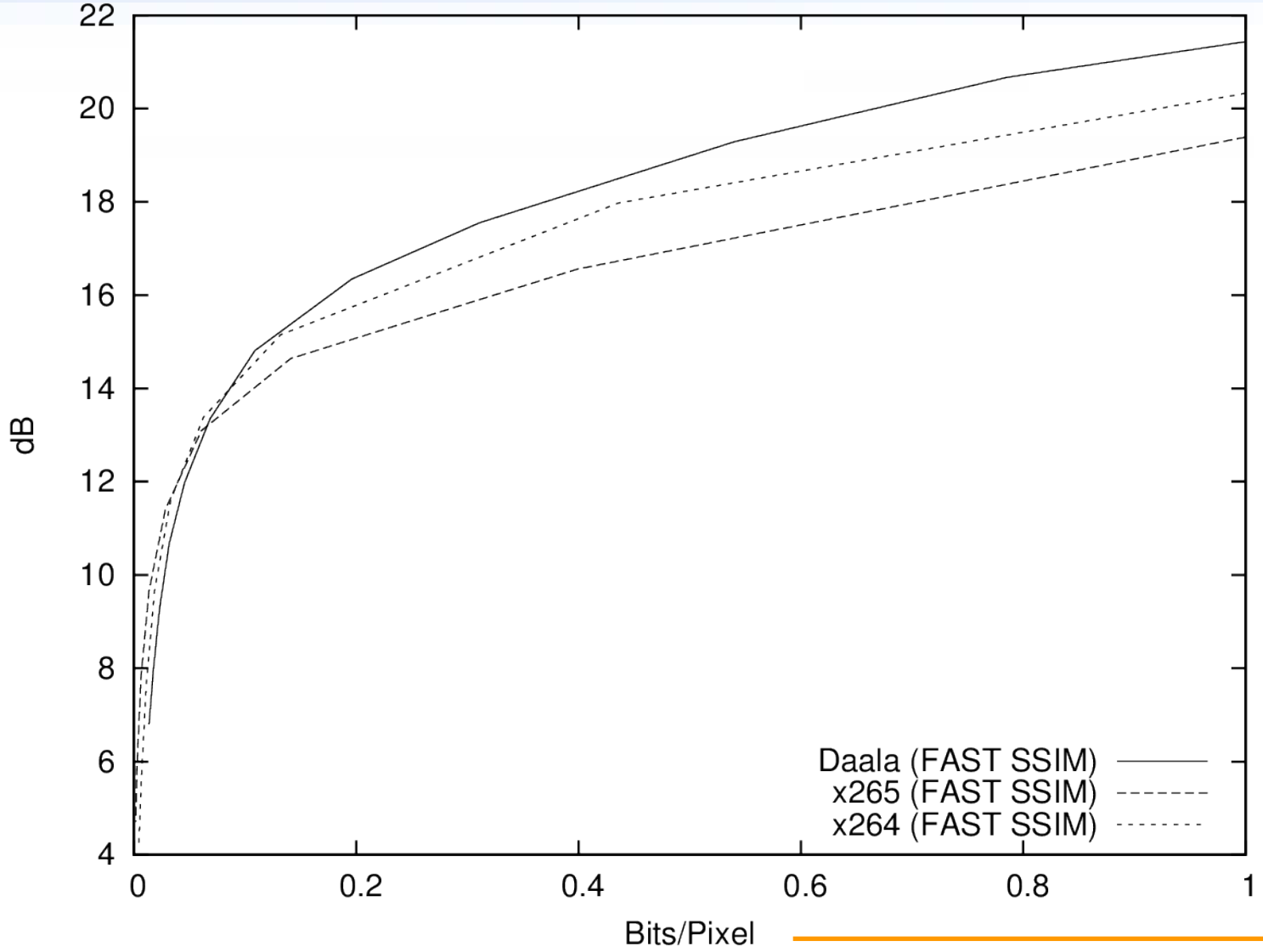
Overall Daala Performance: PSNR



Overall Daala Performance: PSNR-HVS



Overall Daala Performance: Fast[Multi-Scale]SSIM



Resources

- Daala codec website: https://xiph.org/daala/
- Git repository: https://git.xiph.org/
- IRC: #daala channel on irc.freenode.net
- Mailing list: daala@xiph.org

Questions?