The Daala Video Codec: Research Update

Monty Montgomery <*monty@xiph.org*>
(Mozilla, Xiph.Org)

Google VP9 Summit, 2014-06-06

Why Free Codecs Matter

...that's "Free" with a capital F

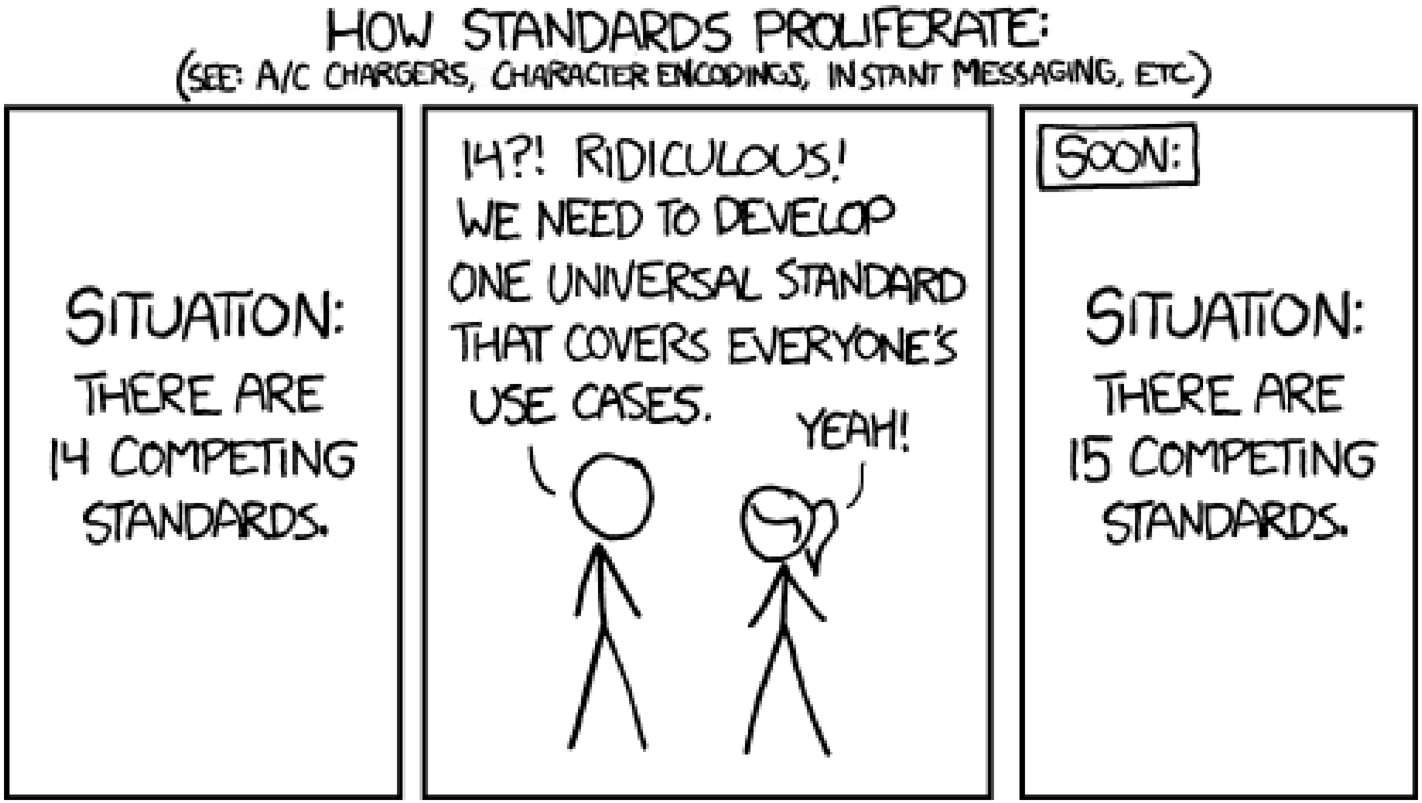
- "Free" refers to control, not [just] cost
- Encumbered codecs are a billion dollar toll-tax on communications tools
- Codec licensing is used as weaponry in competitive battles
  - Licensing regimes are universally discriminatory
- The success of the Internet was based on innovation without asking permission

Why Free Codecs Matter

(continued)

...or begging forgiveness

- Many applications can't tolerate any codec licensing costs at all
  - even the cost of just counting the users is too much
- Ignoring the licensing creates risks that can show up at any time
  - a tax on success
- Compatibility is usually the big cost, not CPU, bandwidth, etc.



http://xkcd.com/927/

...but that's missing the usual motivations behind new codecs!

More and More Codecs

- An organization can't license an encumbered codec when there's no acceptable license offered
- Building a new codec from scratch may cost less than licensing
- Adversarial licensing is a risk in a competitive market
  - FRAND is often none of Fair, Reasonable, or Non-Discriminatory

Changing the Game

- Creating good codecs isn't easy...
  - But we don't need many. Without weird competitive pressures the whole world can cooperate
  - Best implementations of the patented codecs are already often the free software ones
- Where RF is established non-free codecs see no adoption. See: JPEG. Network effect decides
- Unfortunately many different people care about many different things
- Convincing everyone means being better in almost every way, not just one or two

What About Video Codecs?

- Some existing royalty-free formats
  - Theora is circa 1999/2000 technology
  - VP8 solidly better than h264 baseline profile, but the bar was moving to high profile at release
  - VP9 advances performance but shares the same architecture technically and politically
- Structural similarity to patented tech makes FUD too easy
- Single company sponsorship makes some parties uneasy, even with permissive licenses

Strategy is Essential: Pitfalls to Avoid

- Bad IPR story
- Overoptimistic, late rush to market
- Supporting competitors for short term gain
  - and driving off your partners at the same time
- Releasing uncompelling technology
  - Merely competitive isn't good enough when you're the underdog
- Exerting complete control over format
  - Occasionally throwing technology over a wall with a permissive license is not the same as open development.
  - Outside input is needed to improve technology, build an excited community of early adopters, sway critics, and find embarrassing bugs.
  - Giving up all of the above in order to speed time to market isn't worth it.
- Late/Nonexistent hardware support
- A real spec is *not optional*

Strategy is Essential: Now for the DOs

- Design alternatives to avoid the worst patent thickets
- Read and analyze patents, and publish the results
- Patent the new technology we develop
- Use a patent license that encourages adoption and discourages defection
- Target next-next-generation to avoid rushing to market
- Run the open project as an actually open project
- Document, document, document!
  - *“the whole point of a Doomsday Machine is lost if you keep it a secret.”*

Strategy is Essential: These Parts Will Be Hard

- Be best-in-class or go home
- Woo competitors and critics
  - especially those who think they're allies
- Find new niches, uses, applications that are unoccupied and fill them
- Hardware Support

Next Generation Video: Daala

- Lets take some of the strategy that worked in Opus, and apply it to video:
  - Work in a *public process* in a recognized SDO with a *strong IPR disclosure policy* and Opus-like patent licensing
  - Question assumptions in the conventional structure of video codecs, no sacred cows
  - Target applications where high flexibility is essential
  - optimize for *perception* not *PSNR*

30 Second Introduction to Video Coding

Most video codecs use the same basic ideas:

- **Prediction**: Consider what you know about previous or typical content to predict future data
- **Transformation**: Rearrange the information to make it more compressible
- **Quantization**: Strategically lower the resolution of the transformed data
- **Entropy coding**: Code the quantized data taking probability distribution into account

30 Second Introduction to Video Coding: Prediction

- **Intra-Prediction**: Predict portions of the current frame from already decoded portions of the current frame
- **Inter-Prediction**: Predict portions of the current frame from previous decoded frames
  - **Motion Compensation** to eliminate temporal redundancy

Input

⊖

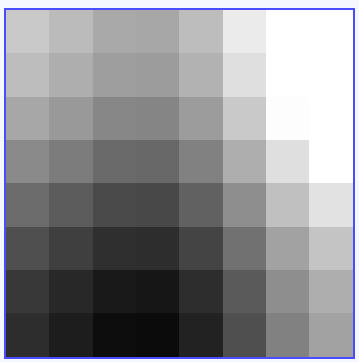
Reference frame

=

Residual

30 Second Introduction to Video Coding: Transformation

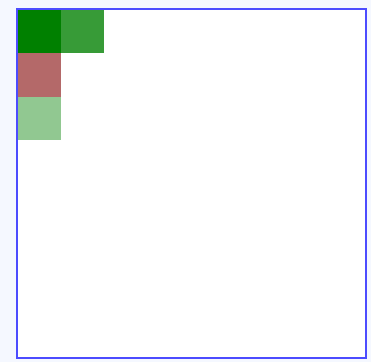
- Map spatial pixel values into some other more compressible representation via a 2D transform, usually the DCT.



Original pixel data

114	108	100	99	109	129	152	166
109	102	95	94	104	124	146	161
99	93	85	84	94	114	137	151
86	80	72	71	82	102	124	138
73	66	58	57	68	88	110	125
60	53	46	45	55	75	97	112
50	43	36	35	45	65	88	102
45	38	31	30	40	60	82	97

DCT



DCT coefficient data

700	200	0	0	0	0	0	0
-150	0	0	0	0	0	0	0
110	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

30 Second Introduction to Video Coding: Quantization and Coding

- **Quantization**: Compute the difference remaining after prediction, then lower its resolution.
  - This is the lossy part
- **Coding**: The quantized error signal is (hopefully) random numbers from some probability distribution.
  - Pack it efficiently into the bitstream

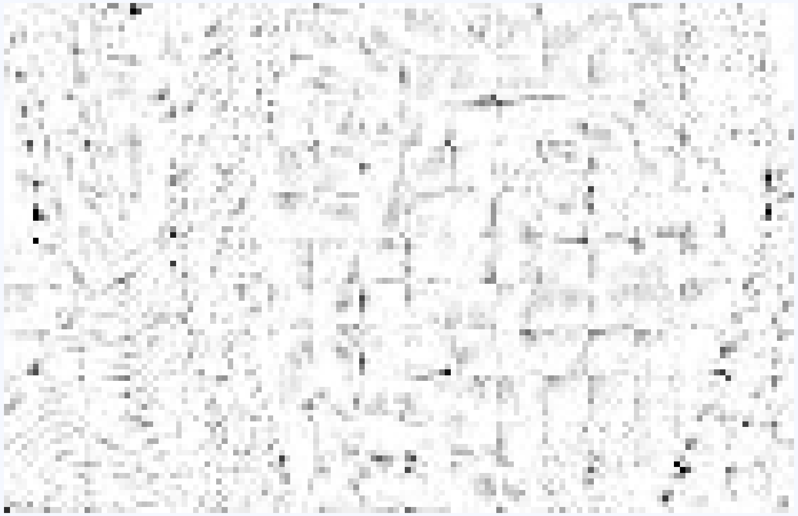
Daala Technological Differences
(so far)

- Lapped transforms rather than traditional DCT
  - Implemented via reversible lifting
- Multisymbol arithmetic encoding
- Frequency domain intra-prediction
- Pspherical vector quantization
- Chroma plane prediction from luma planes
- Overlapping-block motion compensation
- Time-frequency resolution switching

DCT Blocking Artifacts

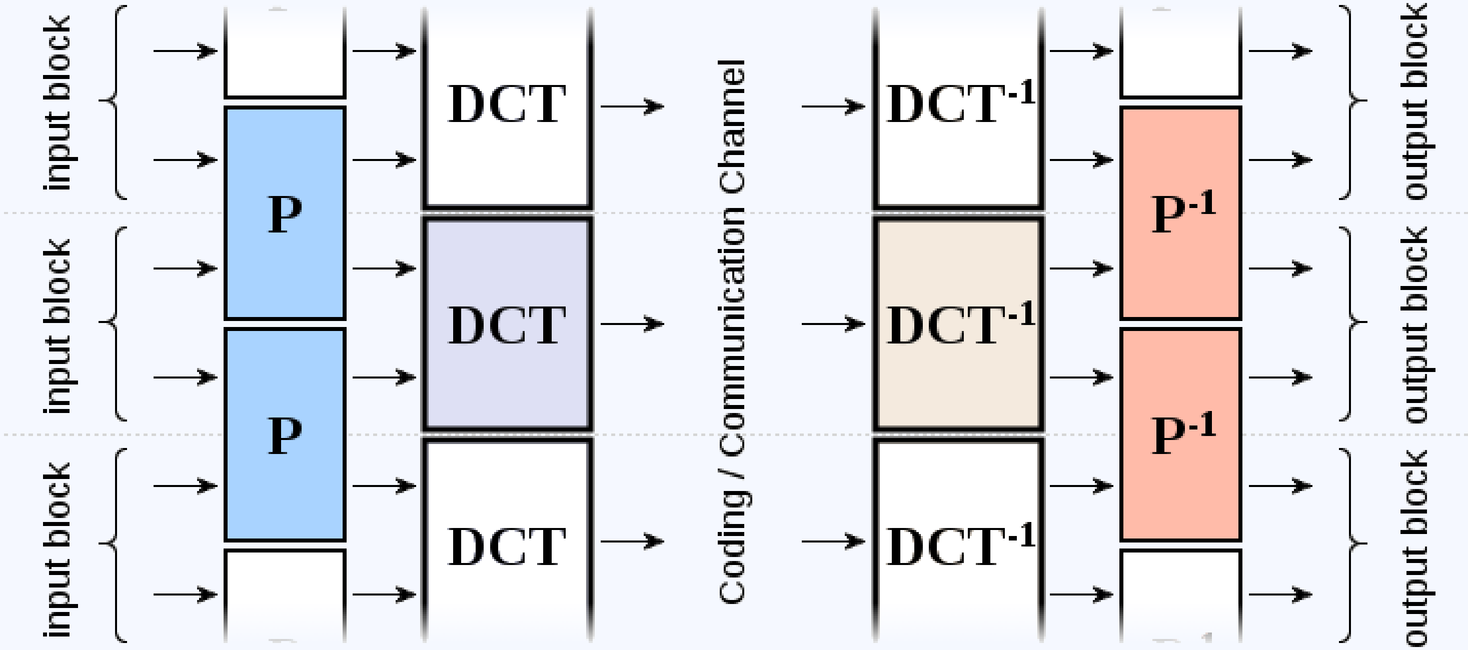
- When we have few bits, quantization errors may cause a step discontinuity between blocks
  - Error correlated along block edge → highly visible
- Standard solution: a “loop” filter
  - Move pixel values near block edges closer to each other

Closeup of reconstructed image



Normalized error distribution within each block

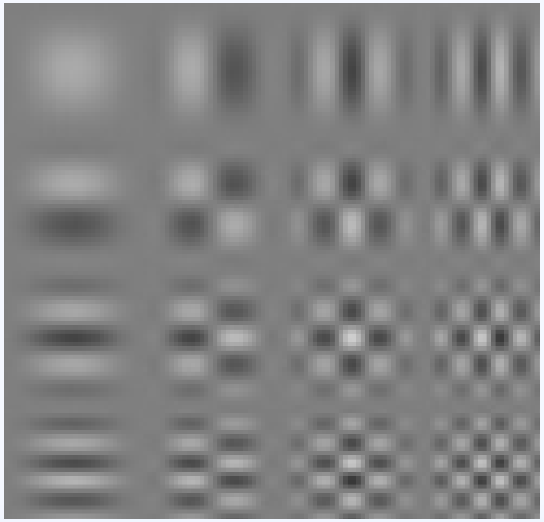
Lapped Transforms



Lapped Transforms

- No more blocking artifacts, without loop filter
- Computationally cheaper than wavelets
- Better compression than DCT or Wavelets
- Doesn't completely disrupt block-based DCT infrastructure
- More details at http://people.xiph.org/~xiphmont/demo/daala/demo1.shtml

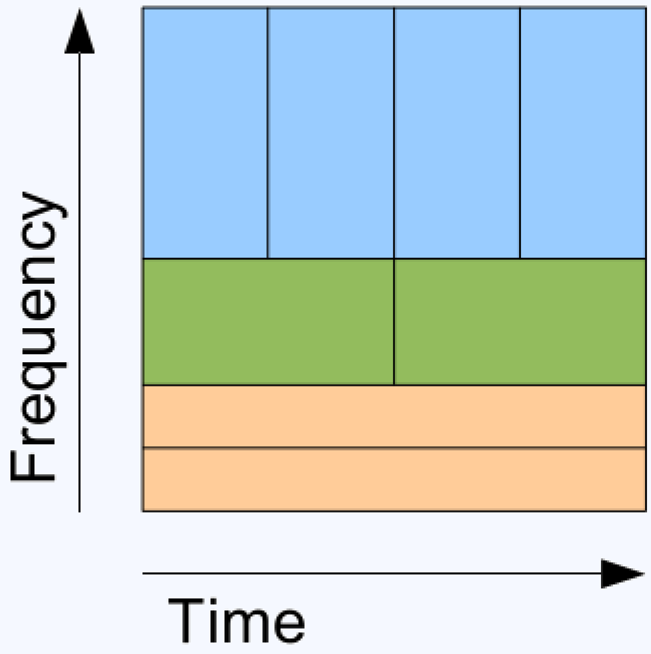
	4-point	8-point	16-point
KLT	7.5825 dB	8.8462 dB	9.4781 dB
DCT	7.5701 dB	8.8259 dB	9.4555 dB
LT	8.6060 dB	9.5572 dB	9.8614 dB
9/7 Wavelet		9.46 dB	



Why not Wavelets?

Wavelets were touted as the next big thing in video coding 10-15 years ago.

- Good LF resolution (models correlation well)
- Better time resolution in HF (prevents ringing)
- Smooth basis functions (no blocking artifacts)



Why not Wavelets?

(continued)

- Good for large scale correlations, but codecs didn't use them for that
- Wavelets break down at low rates
  - HF “texture” requires *more* bits to code separately at every spatial position
  - Extreme low-passing is typical

Original

Dirac @ 67.2 kbps

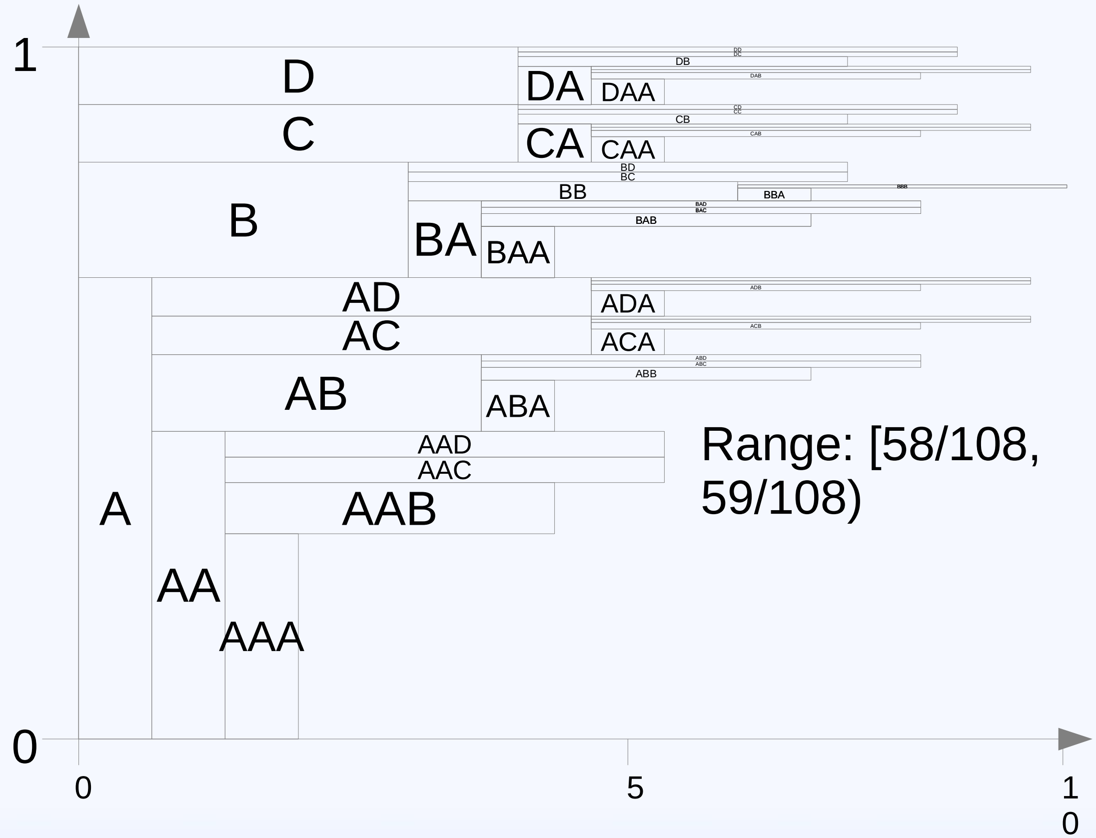
Theora @ 17.8 kbps

Arithmetic Coding

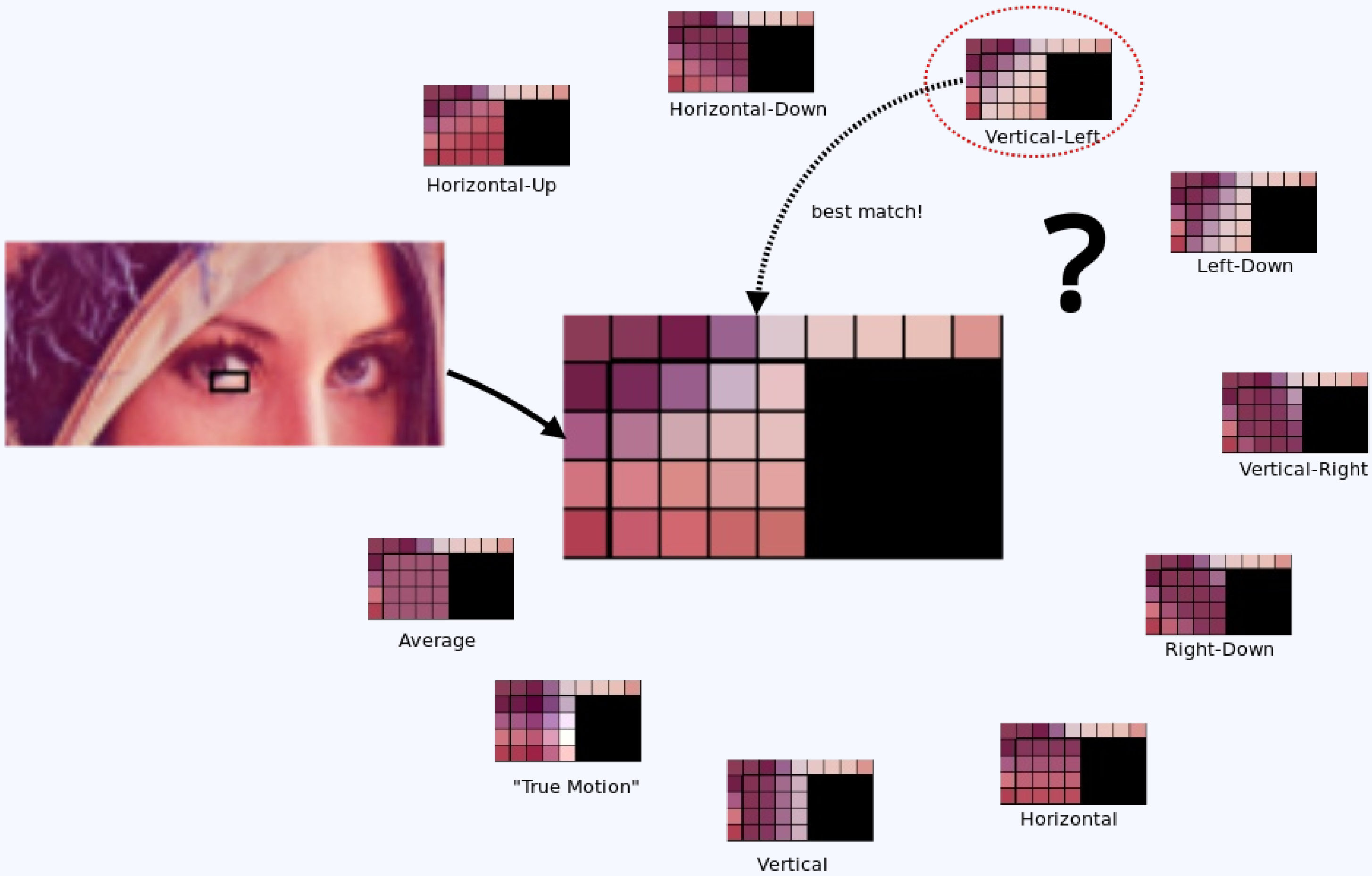
- Given some symbol probabilities, efficiently pack symbols into a bitstream
- Inherently serial; major performance limitation in hardware
- There are many fast approximations if your symbols are binary
  - But many of them are patented
- What about *non-binary*?
  - We used multisymbol coding in Opus

Arithmetic Coding

- Turns out that non-binary coding makes part of the process inherently parallel
- Reduces the serial part in direct proportion to the symbol range
- ~2x speedup when testing on top of VP8
- Multisymbol probability modeling is harder, but often more powerful



Typical Intra-Prediction

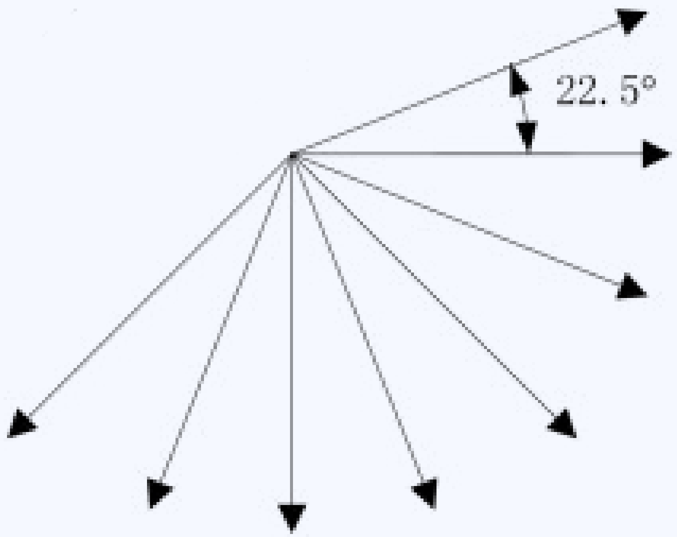


The intra-prediction modes for 4x4 blocks in WebM (VP8).

Typical Intra-Prediction

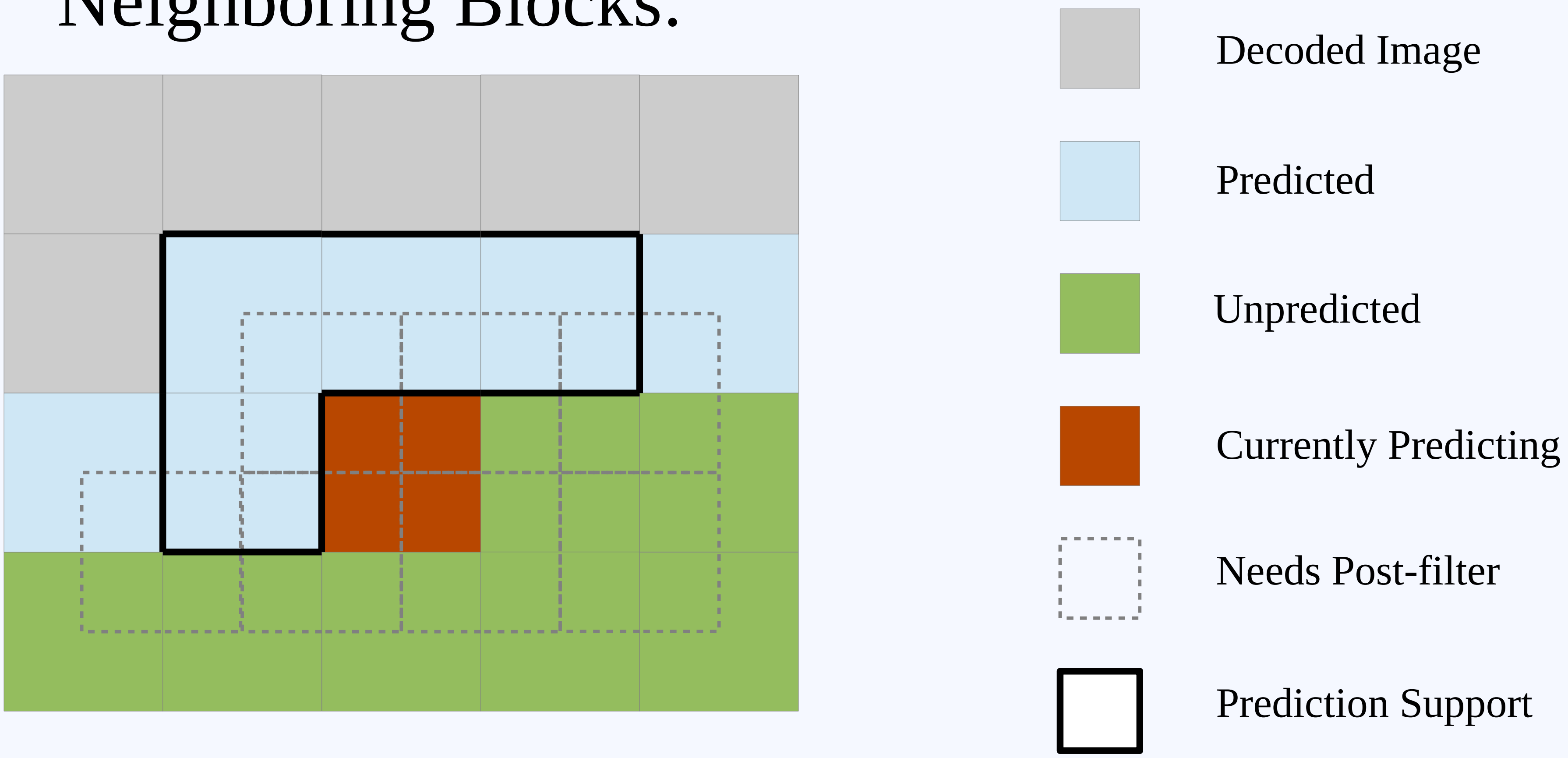
- Pros:
  - Uses image data from neighboring blocks
    - Only need to remember 1 pixel border
  - Parameterizable for any angle θ
  - Predicts difficult to code features well
    - edges are extended
  - Efficient implementation (~no multiplies)

  Cons:
  - Poor prediction in textured areas
  - Blocks L, UL, U, UR must be decoded
    - Doesn't work with overlapped blocks!



Decoding an Intra Frame

Neighboring Blocks:



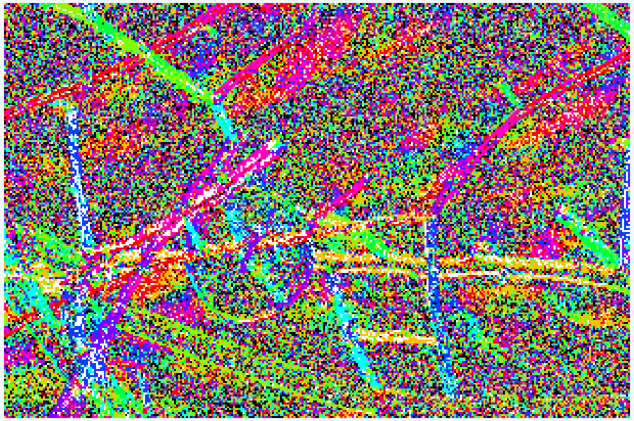
Intra-Prediction in the Coefficient Domain

- We don't have the pixels needed for a traditional intra predictor
- Lapping reduces the need for prediction, but only somewhat
- Why not predict in the lapped DCT domain?
  - Each coefficient for the block predicted as a weighted sum of the neighboring blocks coeffs
  - If not for the lapping we could have the same predictors either way
- “Directions” don't have a clear meaning in the transform domain, so how do we design these?

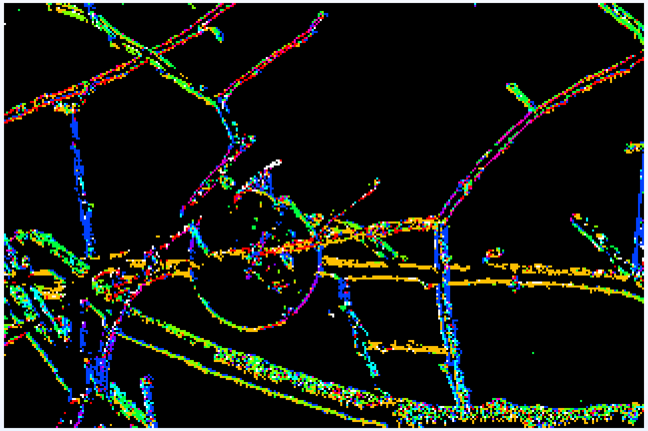
Machine Learning for Intra-Predictors

Training Image

Original VP8 modes



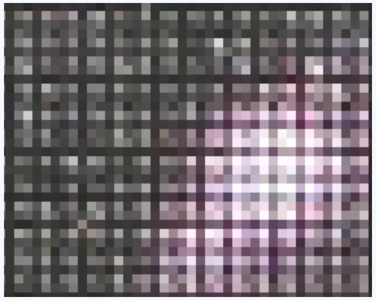
K-Means refinement

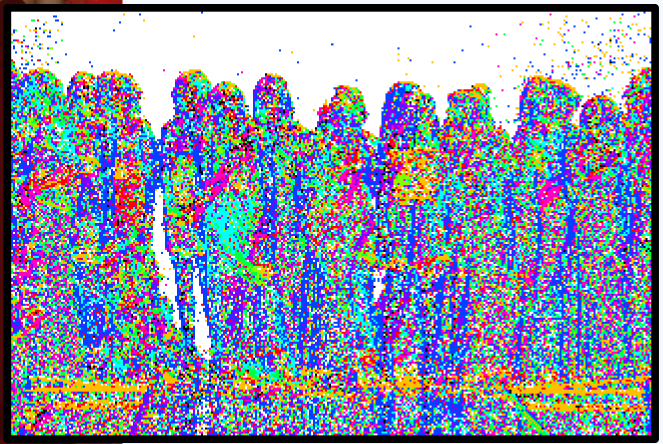


So far: ~0.25 dB more coding gain than classic intrapred, plus actually works with lapped transforms

Not Just Limited to Directions!

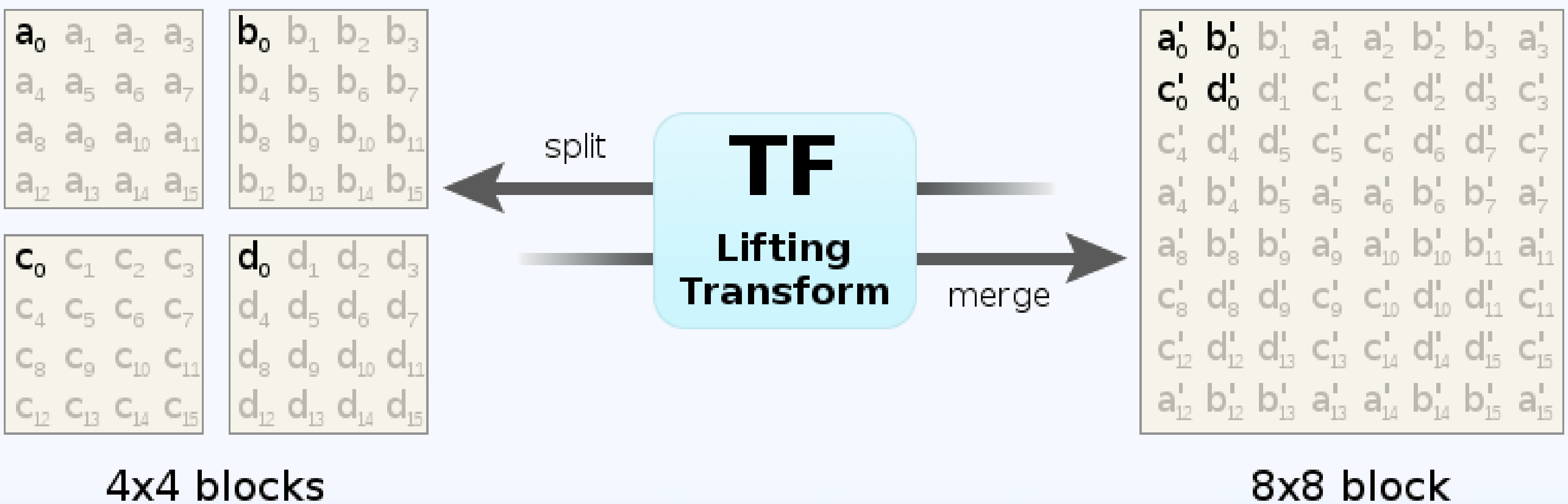
"Mode 1 now predicts periodic texture!"





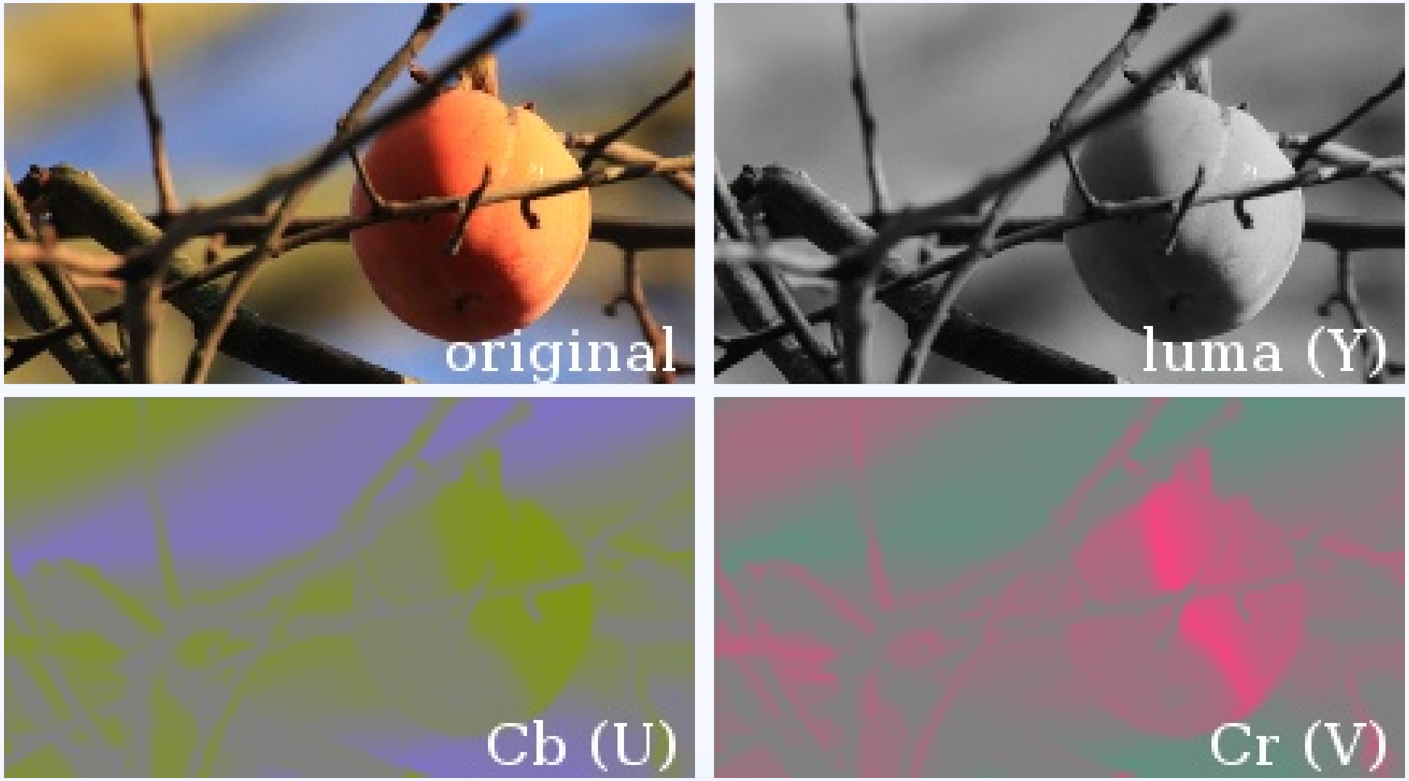
Time-Frequency Resolution Switching (TF)

- Opus uses TF to make different time/frequency resolution tradeoffs in each audio band (thus the name)
- Daala uses TF to cheaply merge/split blocks in the transform domain without reversing or repeating the transform
- More details at http://people.xiph.org/~xiphmont/demo/daala/demo3.shtml



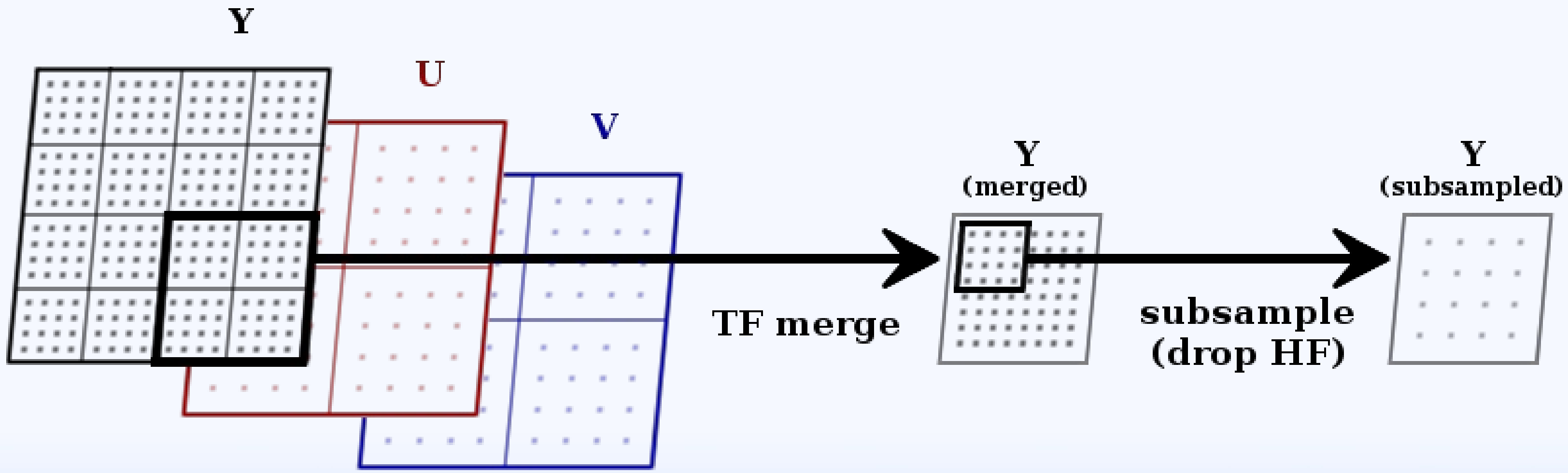
Chroma Plane Prediction from Luma (CfL)

RGB -> YUV moves most of the entropy into the luma channel but residual local correlation remains, esp. edge locations



Chroma Plane Prediction from Luma (CfL)

- Existing published CfL techniques work in the pixel (spatial) domain
- Predicting chroma from luma in the pixel domain can be computationally complex
- But in the frequency domain it's fast!
  - TF enables frequency domain CfL with subsampled chroma

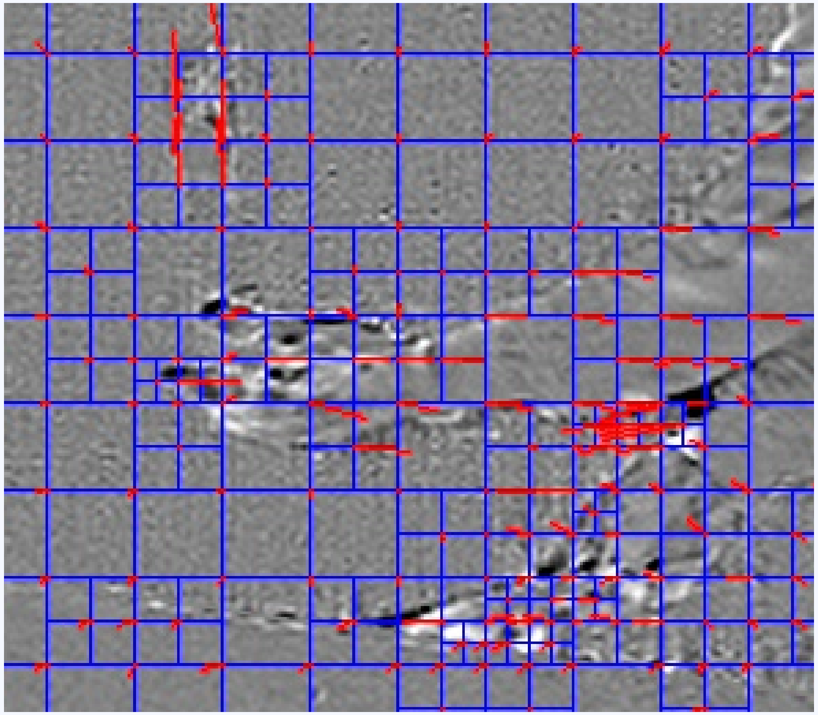


Chroma from Luma
(continued)

- So far, our CfL results do somewhat worse on every objective metric (PSNR, SSIM, fast-SSIM, PSNRHVS)
- But it *looks* clearly better!
  - Most metrics designed on greyscale
- For more information on transform-domain CfL, see: http://people.xiph.org/~xiphmont/demo/daala/demo4.shtml

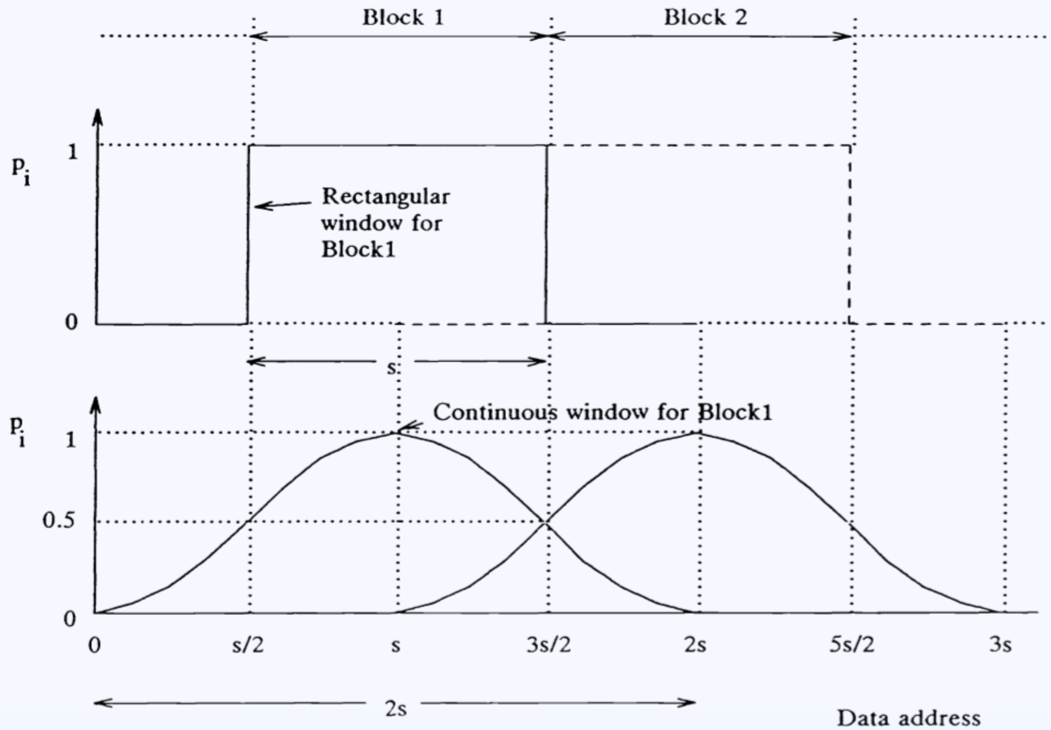
Motion Compensation

- Predict frames from past (sometimes future) frames, compensating for things that move
- Traditional motion compensation displaces blocks of pixels, creates blocking artifacts



Overlapped-Block Motion Compensation

- Overlap the predictions from multiple nearby MVs, and blend them with a window



Also a form of multi-hypothesis prediction

OBMC *(continued)*

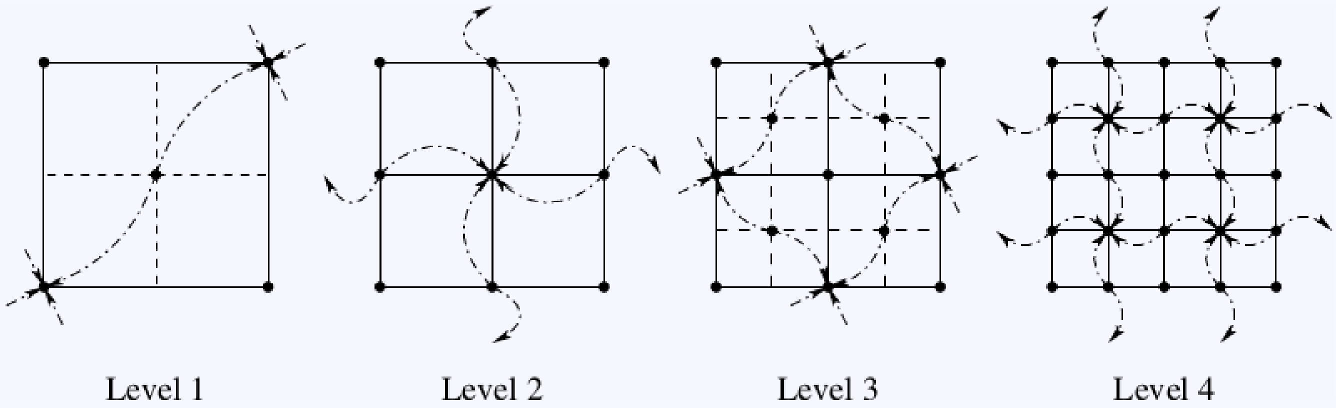
- Used by Dirac
  - Also want to avoid blocking artifacts with wavelets
- PSNR improvements as much as 1 dB
- Issues
  - Motion vectors no longer independent
    - Can use iterative refinement, dynamic programming (Chen and Willson, 2000), bigger cost of ignoring this
  - Can blur sharp features
  - Can add “ghosting” artifacts
  - Handling multiple block sizes

Variable Block Size

- Need a way change block size that doesn’t create blocking artifacts
- Dirac subdivides all blocks to the smallest level and copies MVs
  - Lots of setup overhead for smaller blocks
  - Redundant computations for adjacent blocks with same MV

Adaptive Subdivision

- Allow artifact-free subdivision in a 4-8 mesh
  - Neighbors differ by at most 1 level of subdivision



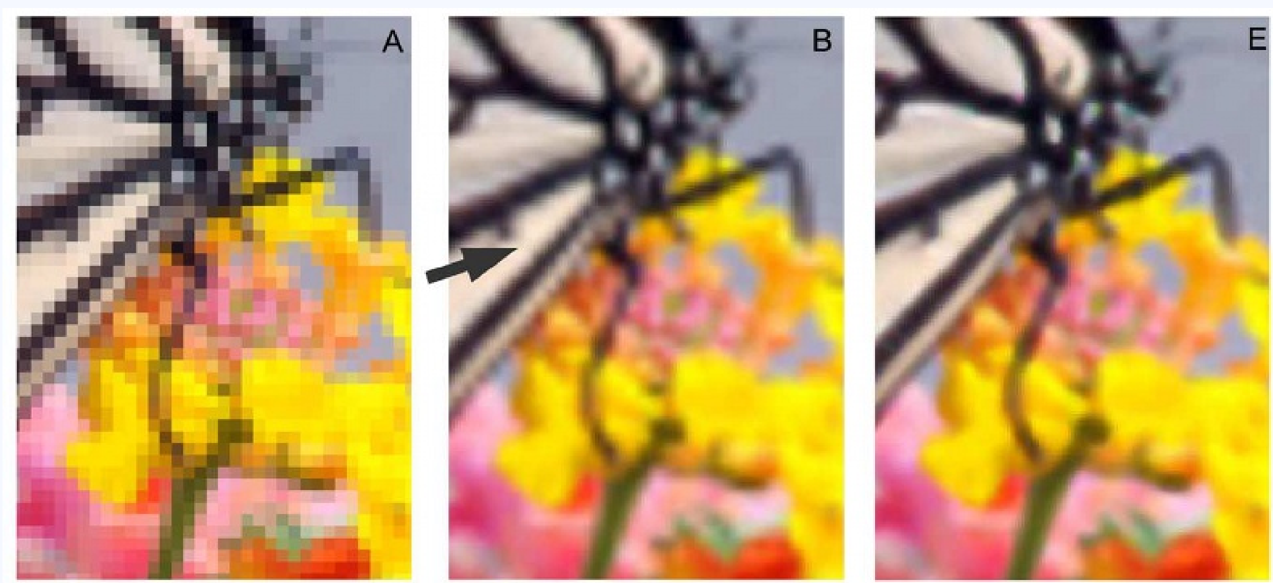
  - Fine-grained control (MV rate doubles each level)
  - Efficient R-D optimization methods (Balmelli 2001)
    - Developed for compressing triangle mesh/terrain data
  - Larger interpolation kernels, less setup overhead, fewer redundant calculations

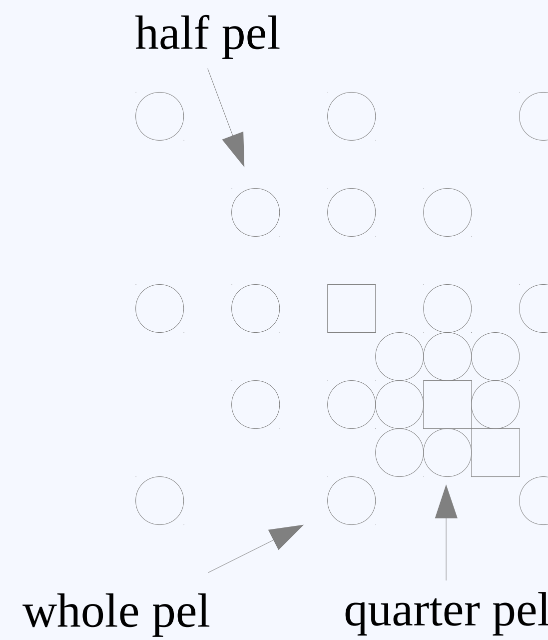
Multiresolution Blending

- Technique due to Burt and Adelson 1983
  - Decompose predictor into low-pass and high-pass subbands LL, HL, LH, HH
  - Blend with small window in high-pass bands
  - Like “enblend” used for panorama stitching
  - Reduces “ghosting” and blurring
- Proposed simplification
  - One level of Haar decomposition (no multiplies)
  - Blend LL band like OBMC, copy the rest
  - Reduces OBMC multiplies by 75%

Edge-directed Subpel Interpolation

- Fractional pixel vectors need interpolation
- Possible to do better than linear filters





- All we need is something fast enough for video

  David Schleef has something, we haven't tried it yet

Pspherical Vector Quantization

- Preserving the overall “energy” in a band turned out to be perceptually critical for audio
  - Opus designed to explicitly preserve energy
- Take a set of values and treat them as a point on an N-dimensional sphere: the radius is the energy, the and the angle is “details”. Code these separately.
- Intuitively it makes sense for image coding: Might it be better for low quality blocks to become “noisy” instead of blurry? “Film grain”
- PVQ provides a convenient, well-tested means of gain-shape coding via algebraic codebooks

Pspherical Vector Quantization
(continued)

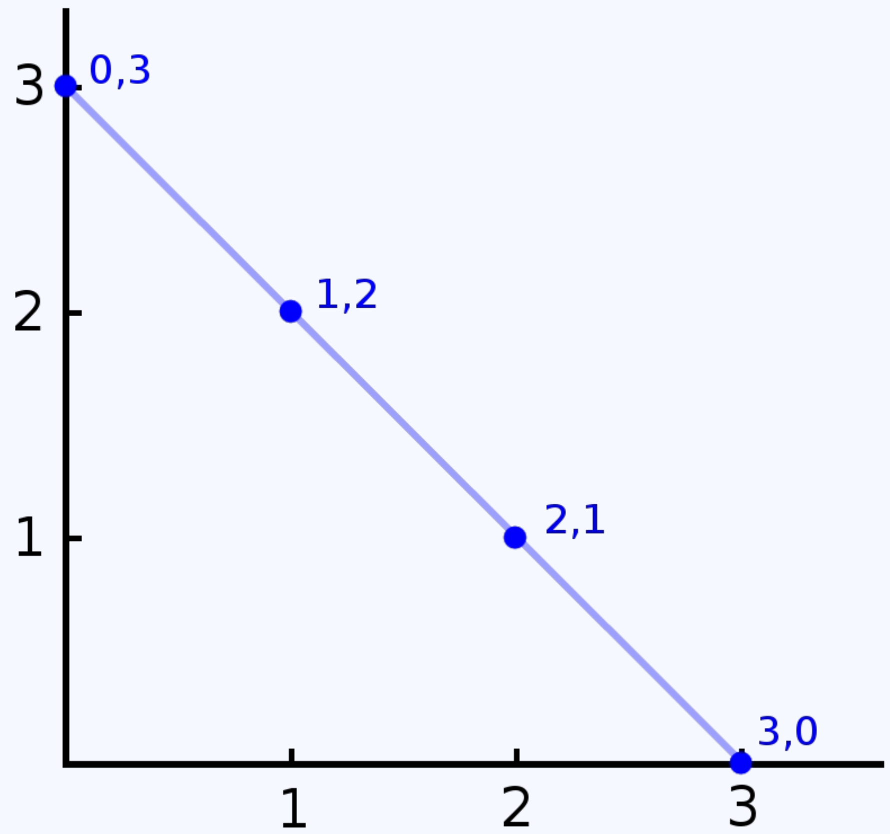
- We want a fast algebraic representation of evenly distributed points on the surface of a sphere
  - Don't know how to do that for arbitrary dimension
  - Use evenly distributed points on a pyramid instead
    - Pyramid Vector Quantization (Fischer, 1986)
  - Warp the *pyramid* into a *sphere*, thus “pspherical”
- For N-dimensional vector, allocate K "pulses"
- Codebook: normalized vectors with integer coordinates whose magnitudes sum to K

$$S(N,K) = \{\frac{\mathbf{y}}{\|\mathbf{y}\|} \in \mathbb{Z}^N : \sum_{i=1}^{N} |y_i| = K\}$$

PVQ Enumeration

Assume the following codebook:

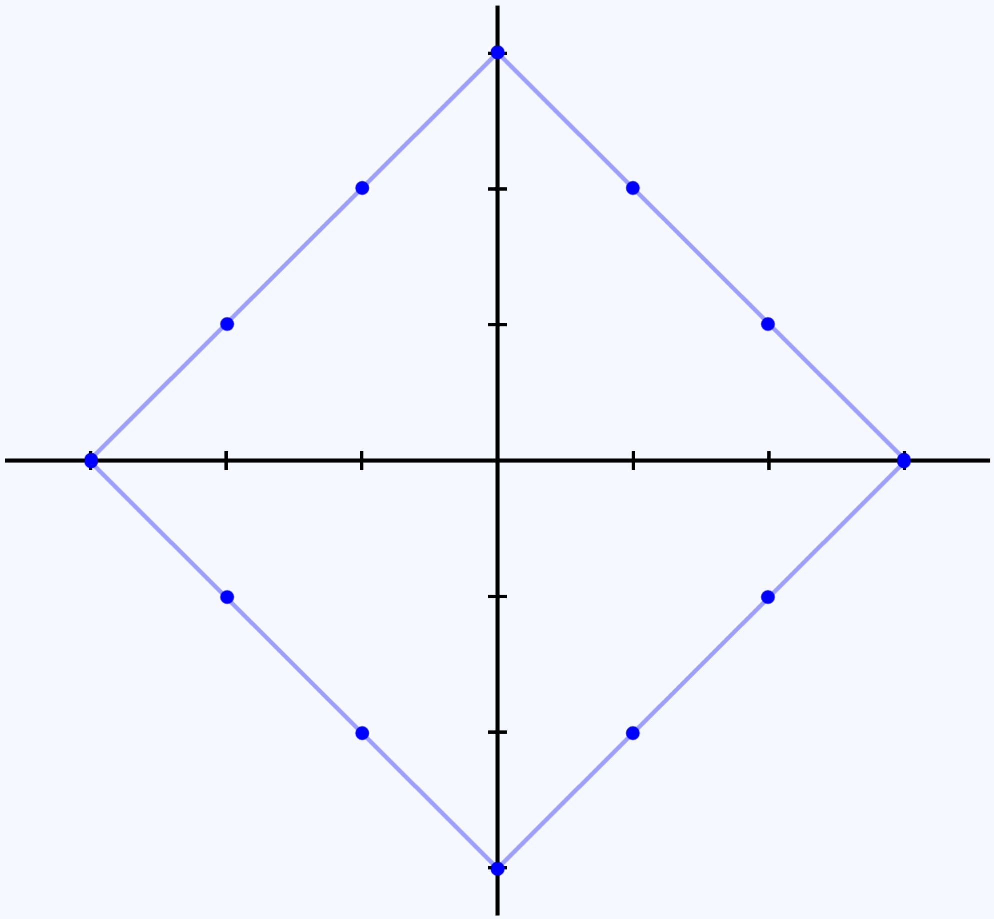
- dimension N=2
- Resolution (pulses) K=3
- Vector values are positive



PVQ Enumeration
(continued)

Assume the following codebook:

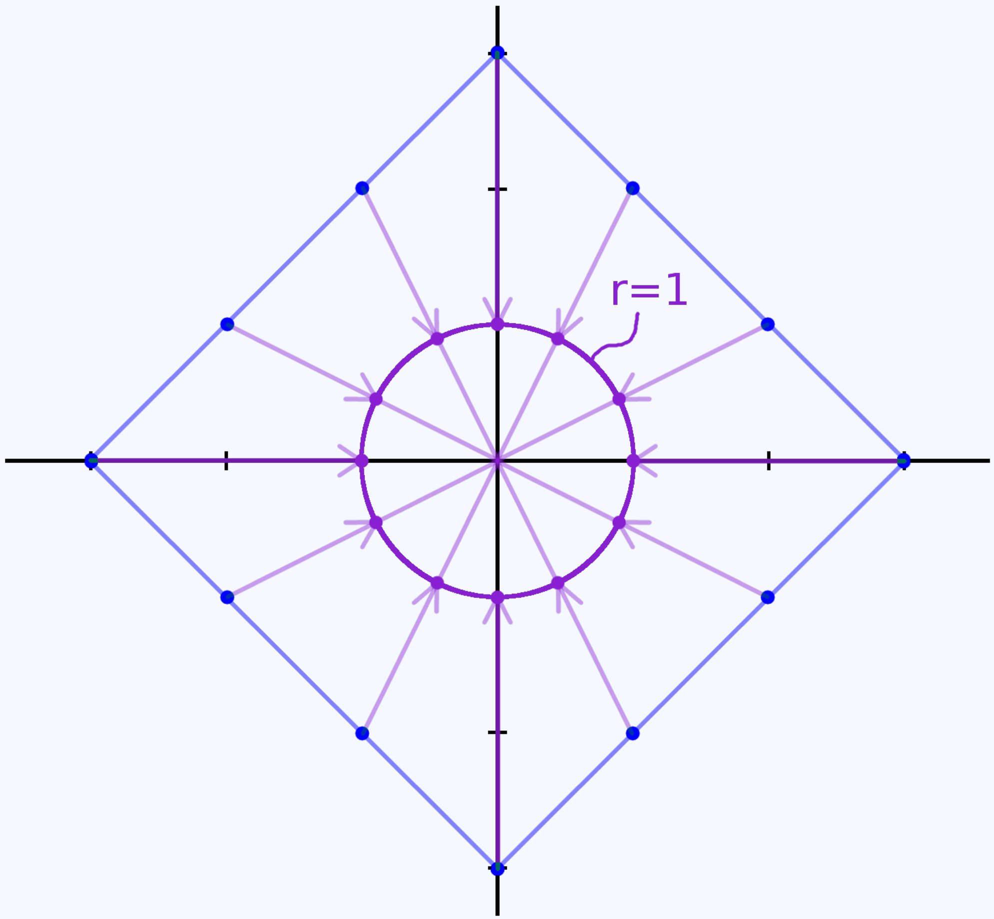
- dimension N=2
- Resolution (pulses) K=3
- Vector values may be positive or negative



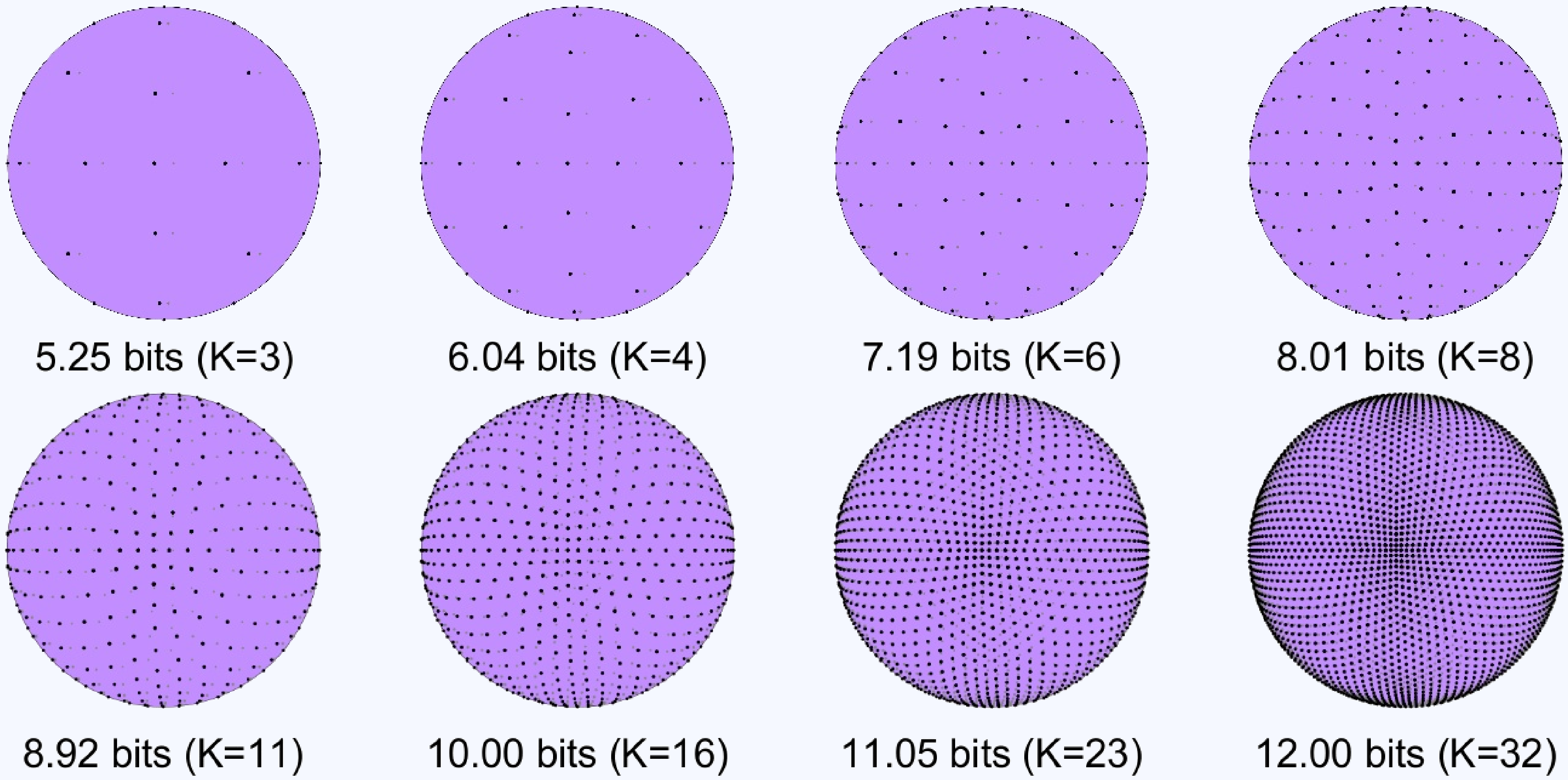
Pspherical Warping

Assume the following codebook:

- dimension N=2
- Resolution (pulses) K=3
- Vector values may be positive or negative
- Project codebook points onto unit circle



Pspherical Codebooks

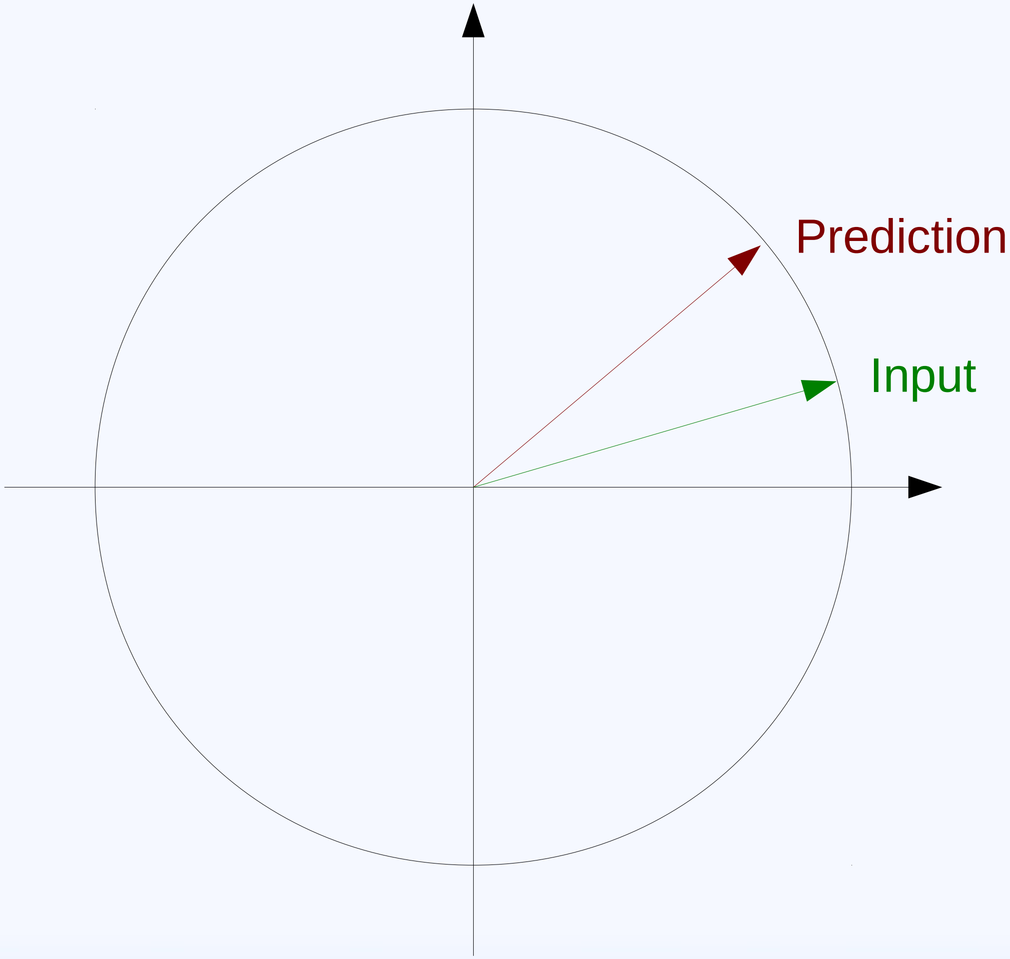


PVQ with Prediction

- Video provides us with useful predictors
- We want to treat vectors in the direction of the prediction as “special”
  - They are much more likely!
- Subtracting and coding the residual would lose energy preservation
- Solution: align the codebook axes with the prediction, treat one dimension differently

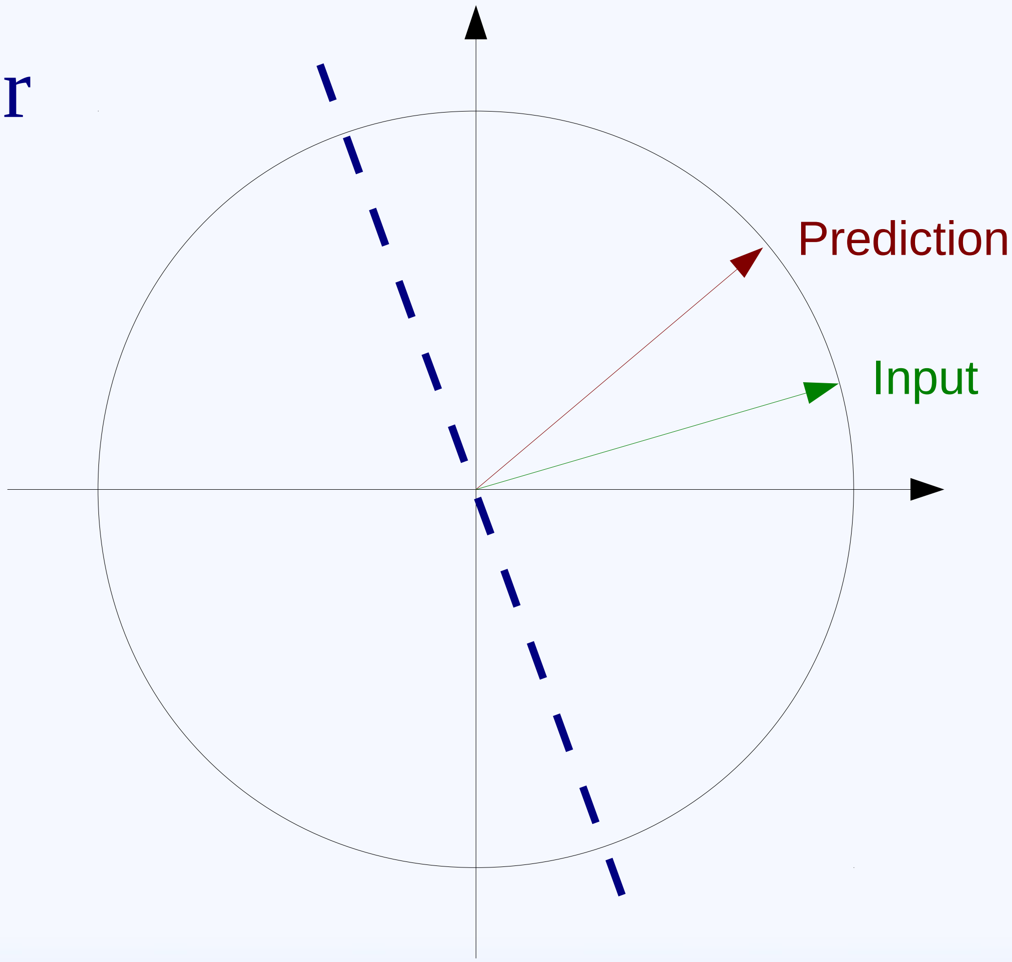
2-D Projection Example

- Input + Prediction



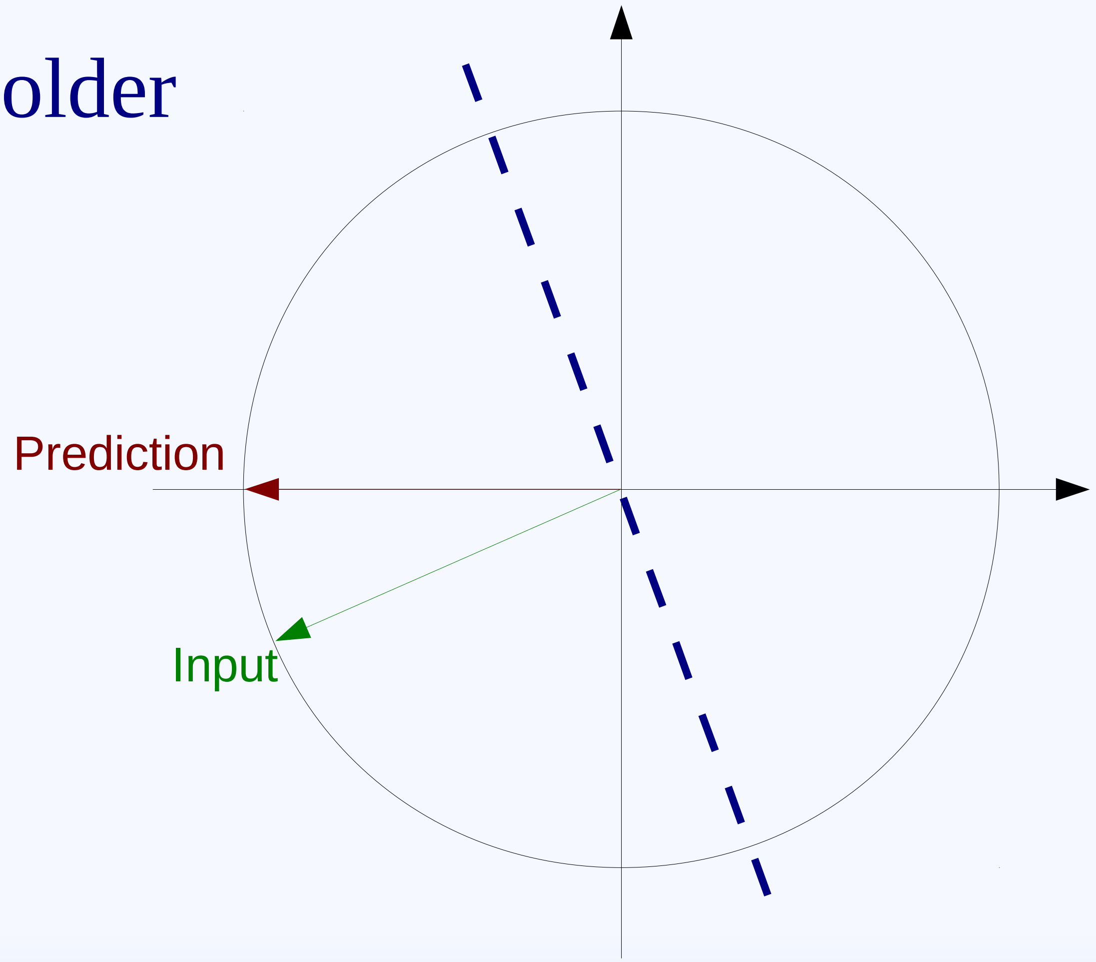
2-D Projection Example

- Input + Prediction
- Compute Householder Reflection



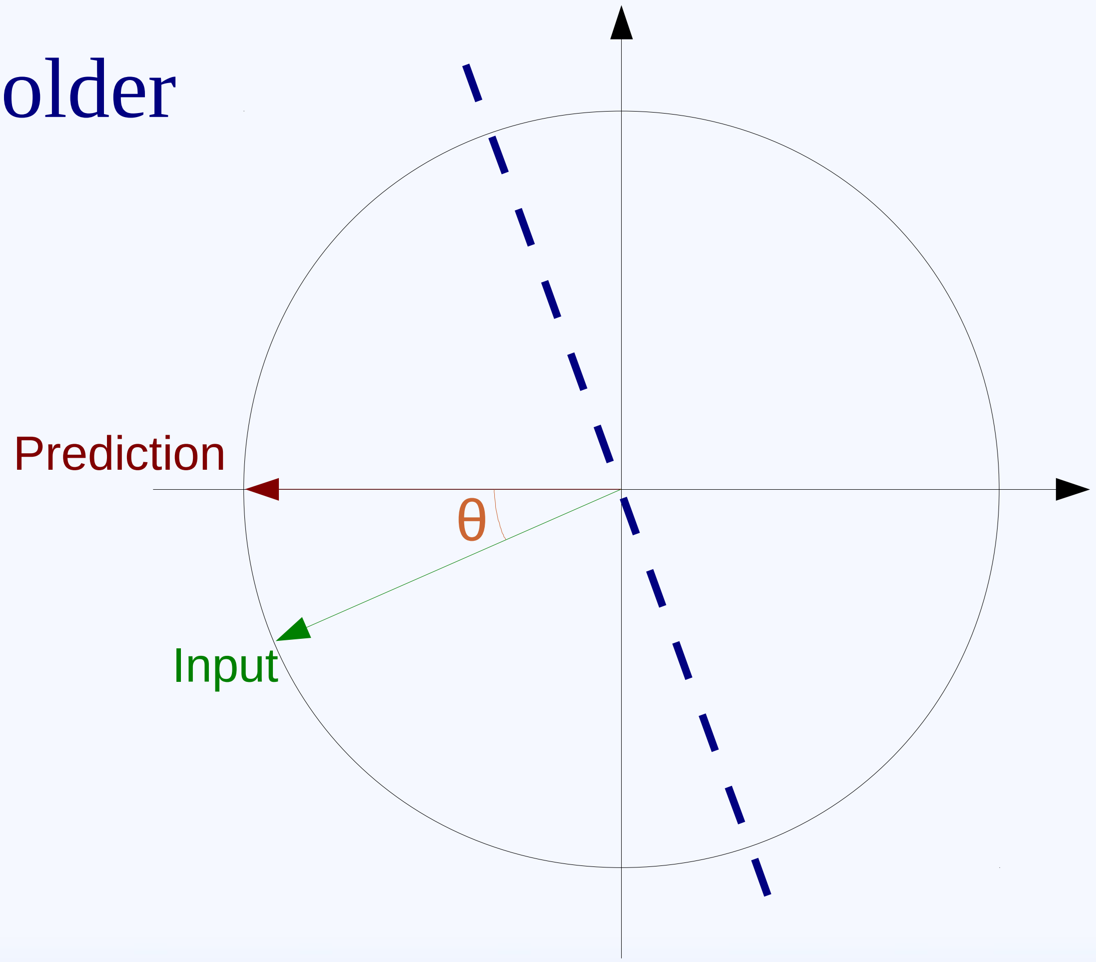
2-D Projection Example

- Input + Prediction
- Compute Householder Reflection
- Apply Reflection



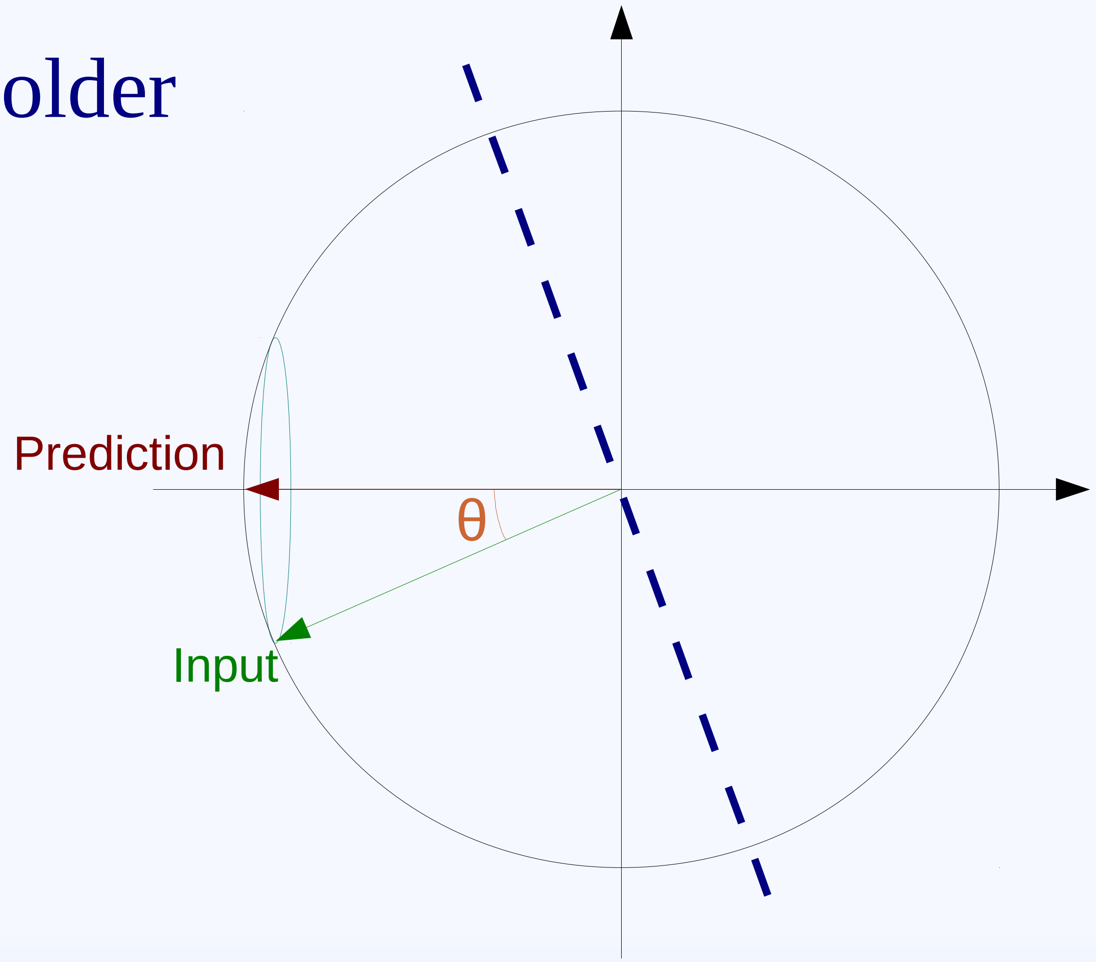
2-D Projection Example

- Input + Prediction
- Compute Householder Reflection
- Apply Reflection
- Compute & code angle



2-D Projection Example

- Input + Prediction
- Compute Householder Reflection
- Apply Reflection
- Compute & code angle
- Code other dimensions



Prediction via Theta-PVQ

- Creates another “intuitive” parameter, θ
  - “How much like the predictor are we?”
  - $\theta = 0 \rightarrow$ use predictor exactly
- θ determines how many pulses go in the “prediction” direction
  - K (and thus bitrate) for remaining N-1 dimensions adjusted down
- Remaining N-1 dimensions have N-2 degrees of freedom (no redundancy)
  - Can repeat for more predictors

Today's Formats Are a Long Way From Exhausting the Possible

How about unblending a cross-fade?

Spatial Sparsity-Induced Prediction for Images and Video: A Simple Way to Reject Structured Interference
Gang Hua and Onur G. Guleryuz (2011)

Recent Work / Updates

Monty's demo pages at:
https://people.xiph.org/~xiphmont/demo
document and explain many of these techniques in more detail, but there have been new developments even since then.

Demo 5 (discussing Pspherical Vector Quantization in detail) is coming, I promise!

The Road Ahead

- The techniques we've been working with appear to work, but there is much to be done
- Industry is currently distracted figuring out how they're going to deploy HEVC (/VP9)
- Your participation is welcome!
  - http://xiph.org/daala
- Opus benefited from some applications served by no other audio codec.
  - Does something similar exist for video?

Daala: Additional Resources

- Website: http://www.xiph.org/daala
- Mailing list: daala@xiph.org
- IRC: #daala on irc.freenode.net
- Git repository: git://git.xiph.org/daala.git
- Demos: http://people.xiph.org/~xiphmont/demo/

Questions?