# How do we justify ALMA-64?

Mark Holdaway
President, Kalimba Magic
**www.KalimbaMagic.com**

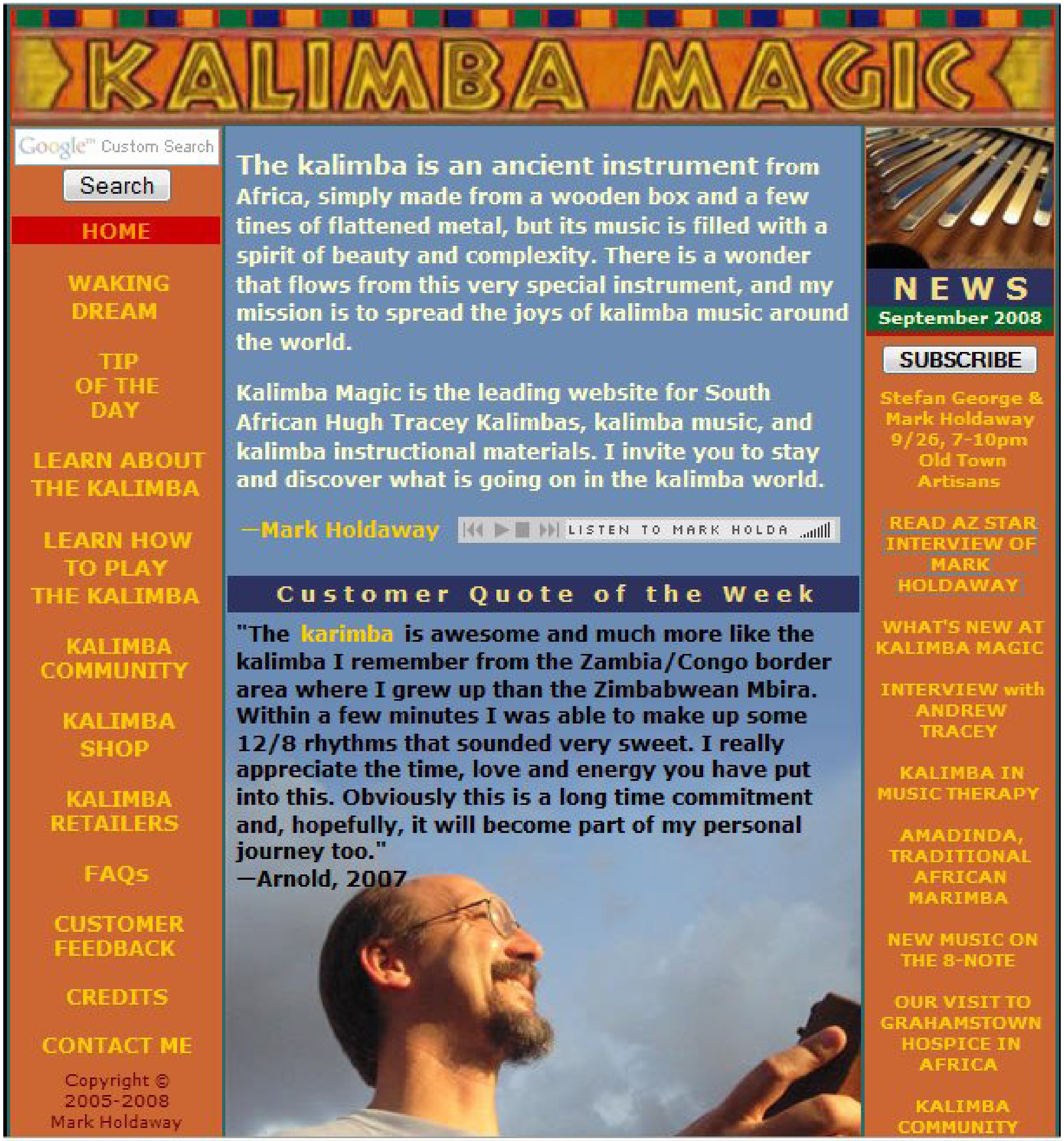
# How do we justify ALMA-64?

Mark Holdaway
President, Kalimba Magic
**www.KalimbaMagic.com**

**NRAO 1989-2007** - focus on MMA/ALMA simulations, imaging, and calibration

**Next 18 years** - sell One Million Kalimbas, create 100 jobs in Africa, start Giving Back to Africa non-profit. 2008: 2000 kalimbas, 60% growth

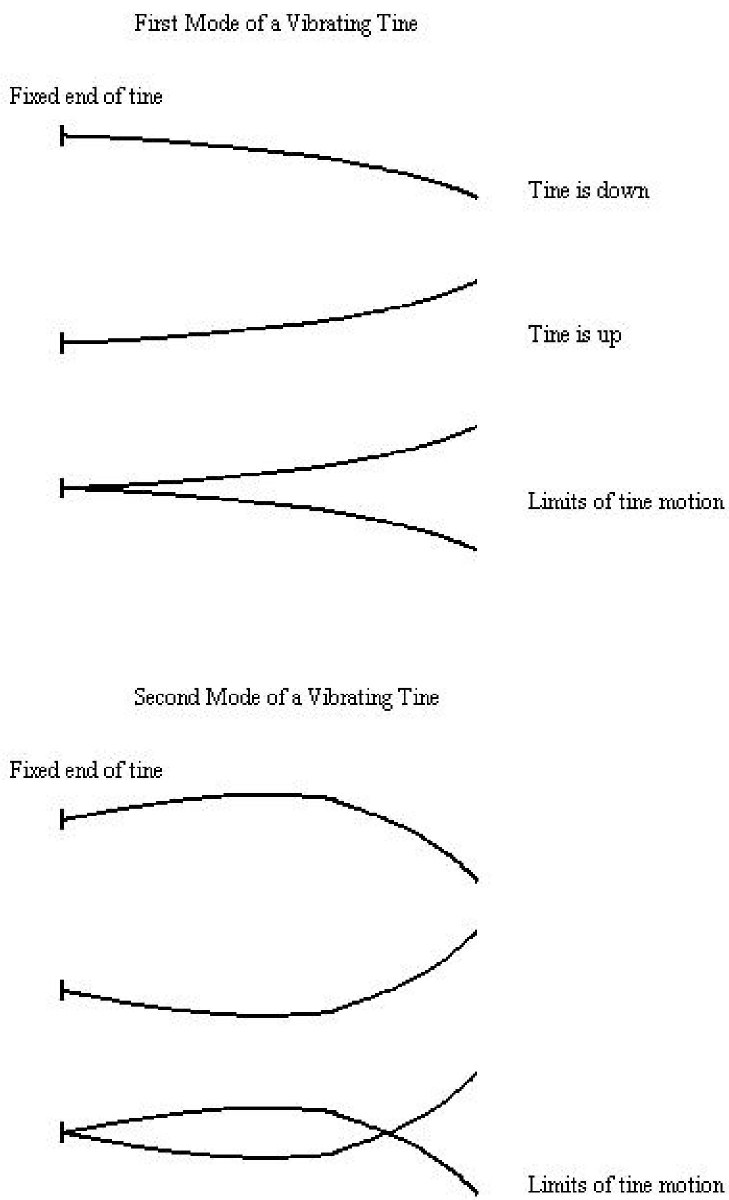
**As self-employed ALMA pundit with no accountability,** I'm free to say anything. Hopefully some is true and useful.



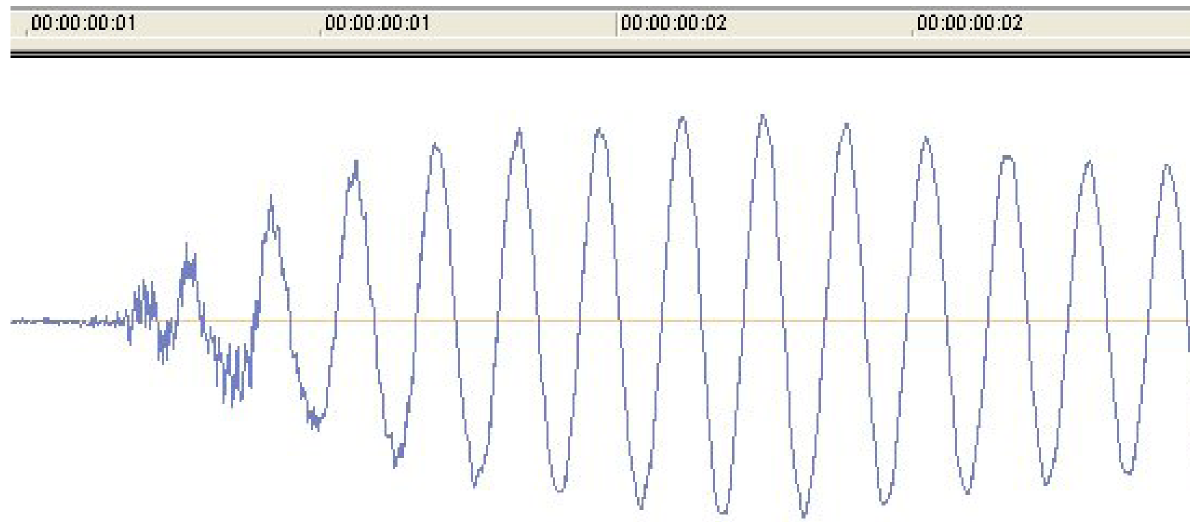

I am still using all of the skills I honed working on ALMA:

Web Design

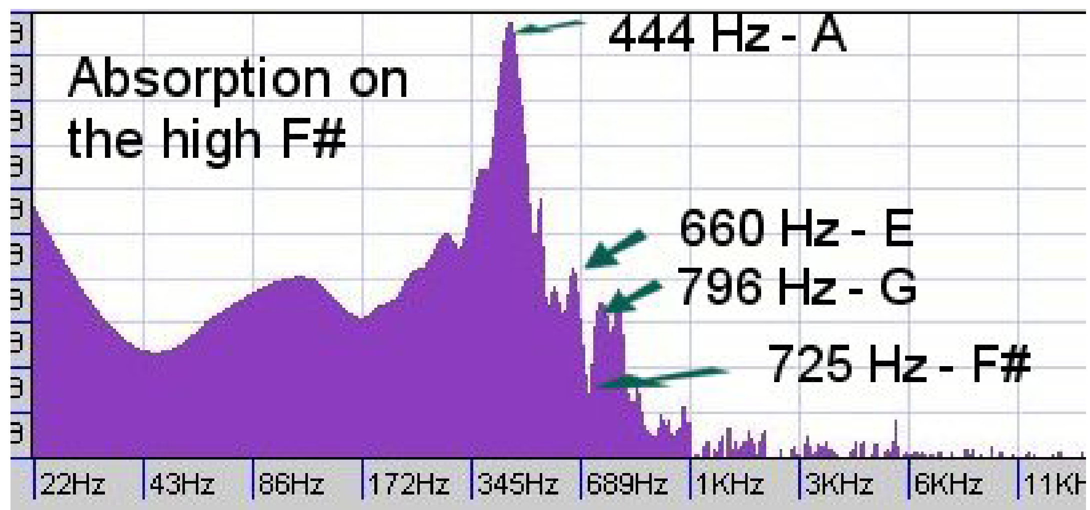
# Vibrational Analysis



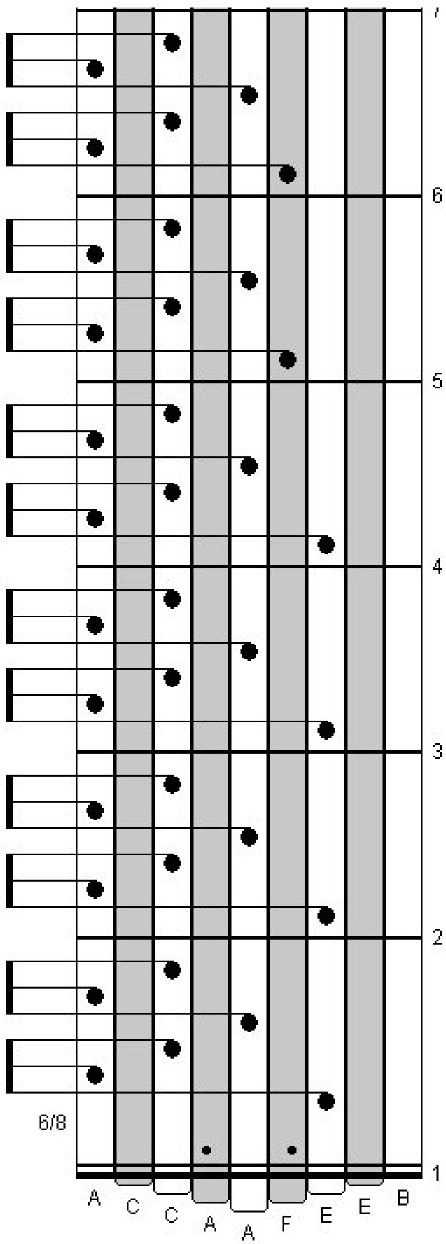

# Time Series Analysis



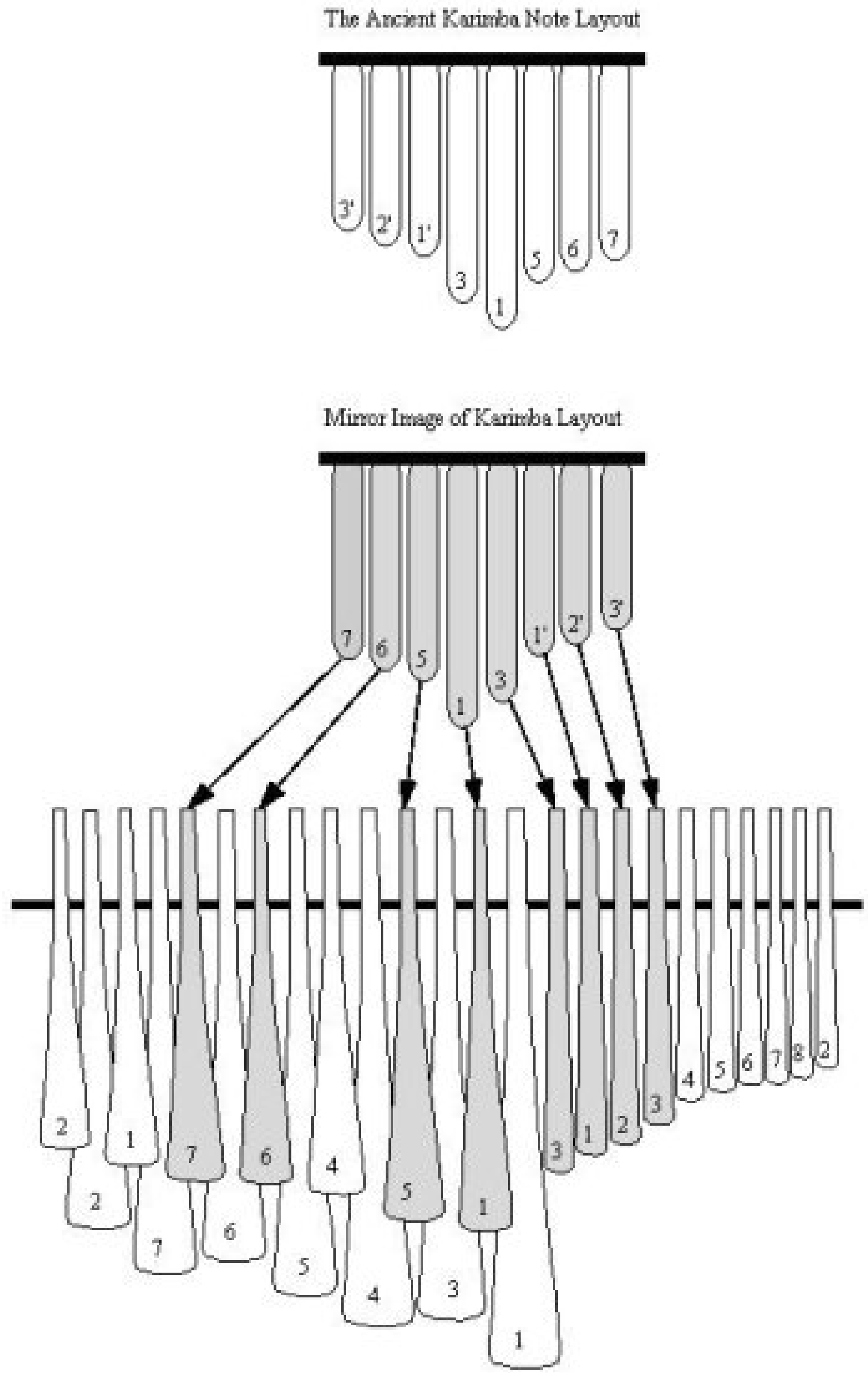

# Spectral Analysis



# Kalimba "Simulations"




KTabS =
Kalimba
Tablature
Software
Design



# Anyway.......
# We have ALMA-50, we want ALMA-64.
# How do we get it?

**All gains are incremental gains**

**Can the sum of all incremental gains add up to something significant?**

**My guess: going to 64 antennas may significantly increase/improve sub-millimeter observation time, and modestly improve all ALMA functionality.**

# What are the incremental improvements?

- Sensitivity Gain: 64/50 = 1.28   OK, thats nice.

- Time Gain: we can track time changes to the same
  senstivity faster by (64/50)^2 = 1.64

  Very fast mm variability is rare (solar flares?),
  OR this gain could be used in a more general way,
  tracking cal parameters to be applied to observations.

  HOWEVER, with higher sensitivity on the target source,
  we also need more accuracy on the cal parameters.

# What are the Incremental Improvements?

- (u,v) Coverage Gain: roughly $(64/50)^2$ more samples

  HOWEVER: imaging simulations of a close protoplanetary disk ended up being noise limited!

  Small, simple, and weak objects won't benefit from the gain in (u,v) coverage.

  So, we need to observe large, complex, bright objects: Planets? Nearby Galaxies? Bright HII Regions? Small Configurations, Low Spectral Resolution?

  ALSO, image quality of such objects may be limited by deconvolution algorithms - need things like NNLS

# What are the Incremental Improvements?

- There is a (u,v) coverage / speed issue:

  For single field imaging, finite support permits good imaging with only partial (u,v) coverage.

  In mosaicing, emission fills the beam, and you generally need “complete (u,v) coverage”.

  For compact arrays, you get complete (u,v) coverage in a single snapshop.

  Larger arrays will require some earth rotation synthesis to get complete coverage.The max config size for complete snapshop coverage scales with N.

# What are the Incremental Improvements?

- There is an incremental improvement in Self-Calibration:

  Traditional phase (or amp + phase) self-calibration

  Pointing sel-calibration

# How N=64 Can Help in Self-Calibration

Self-Calibration is iteratively, alternatively solving for the image and calibration parameters.

- Better (u,v) coverage => Better inherent imaging (sometimes)

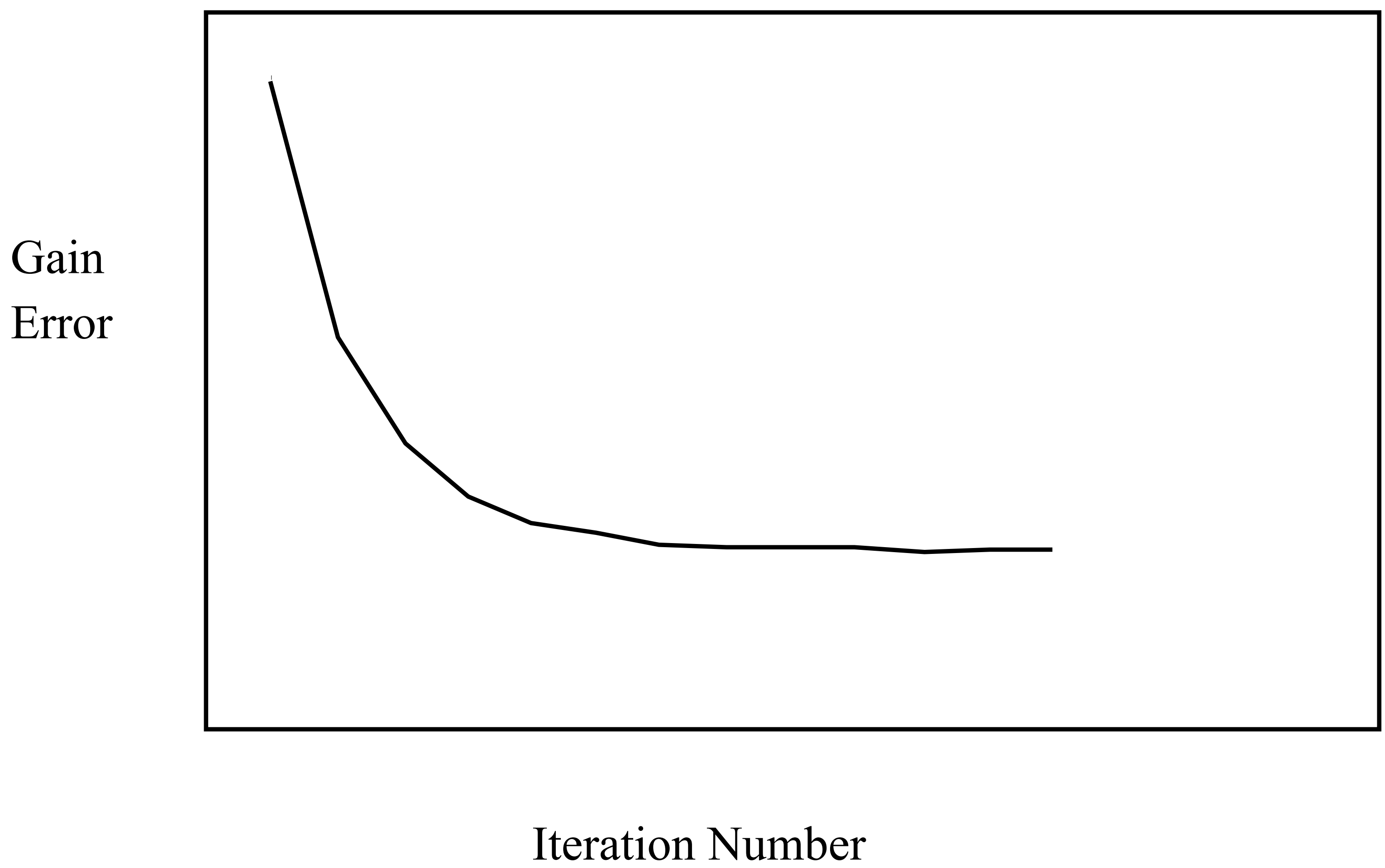
- (u,v) redundancy results in some cancelation of errors.

- Better imaging => Better model input for self-cal loop

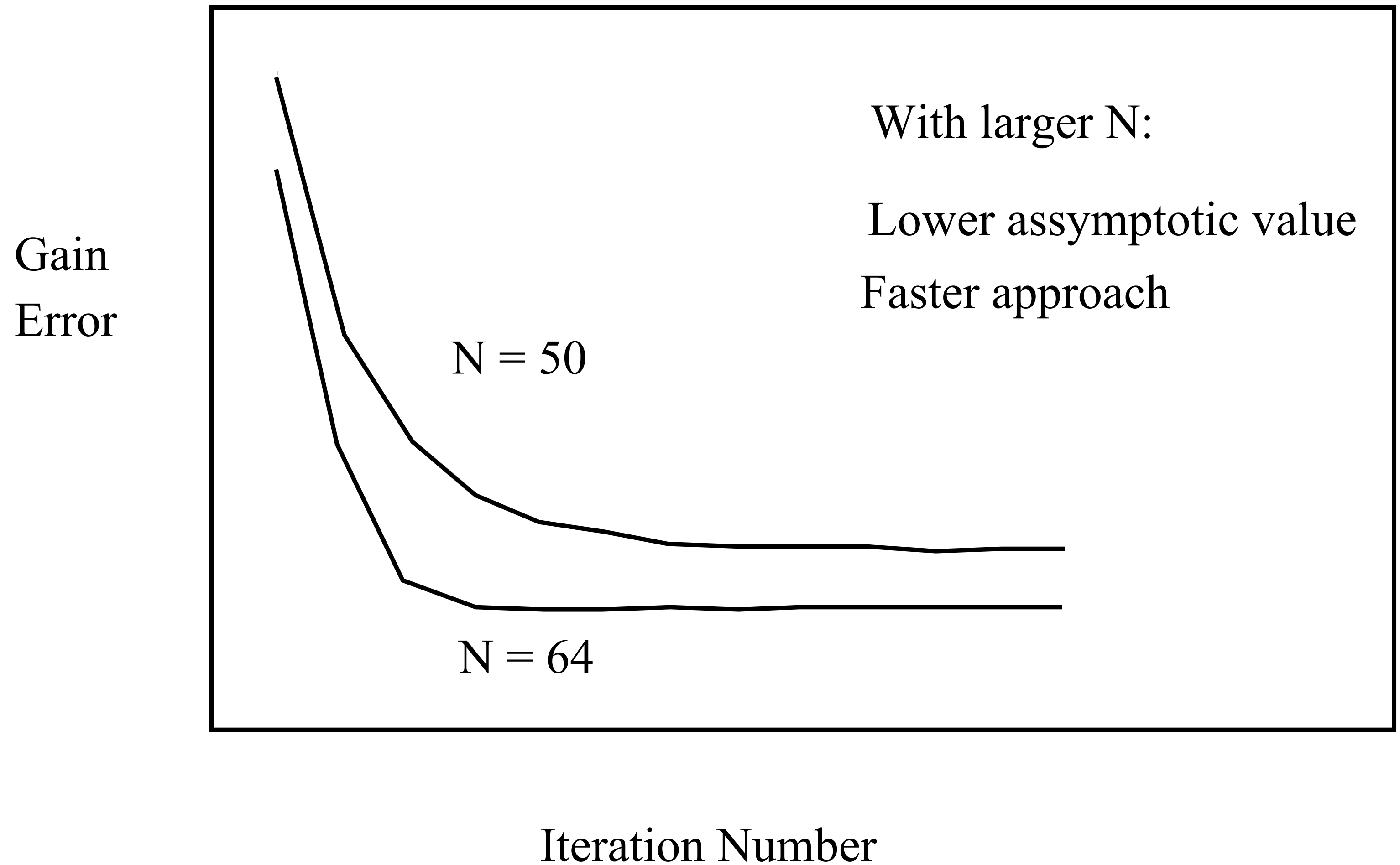
- Larger N => higher sensitivity in gain solutions.
  Gain errors go like 1/N

- Sometimes, atmospheric fluctuations are faster than the time required to detect the gains with sufficient sensitivity.

# How N=64 Can Help in Self-Calibration



# How N=64 Can Help in Self-Calibration

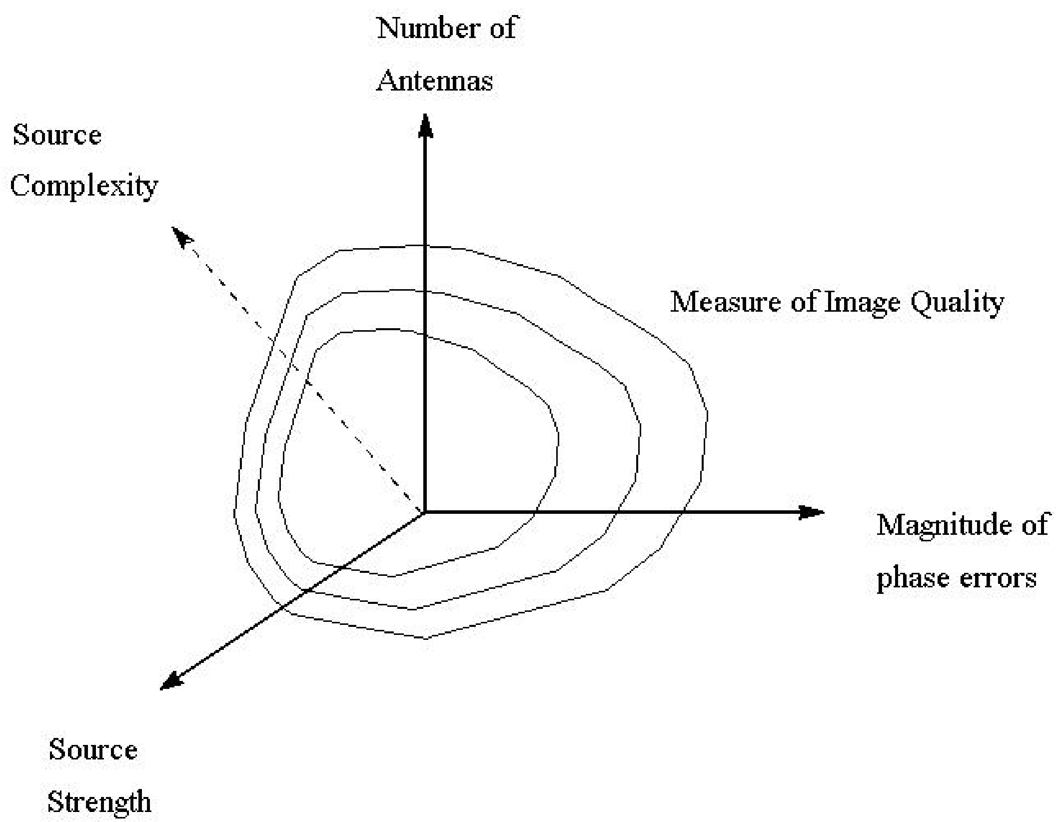



SO: for the same number of self-cal iterations, you get to a smaller residual error; OR, you get to the same error level with fewer iterations and less effort.

# How I See Simulations:

sampling a region of multi-dimensional phase space

| Dimensions: | | |
|---|---|---|
| | Nants | discrete |
| | Phase error magnitude | continuous |
| | Source strength | continuous |
| | Source complexity | discrete |

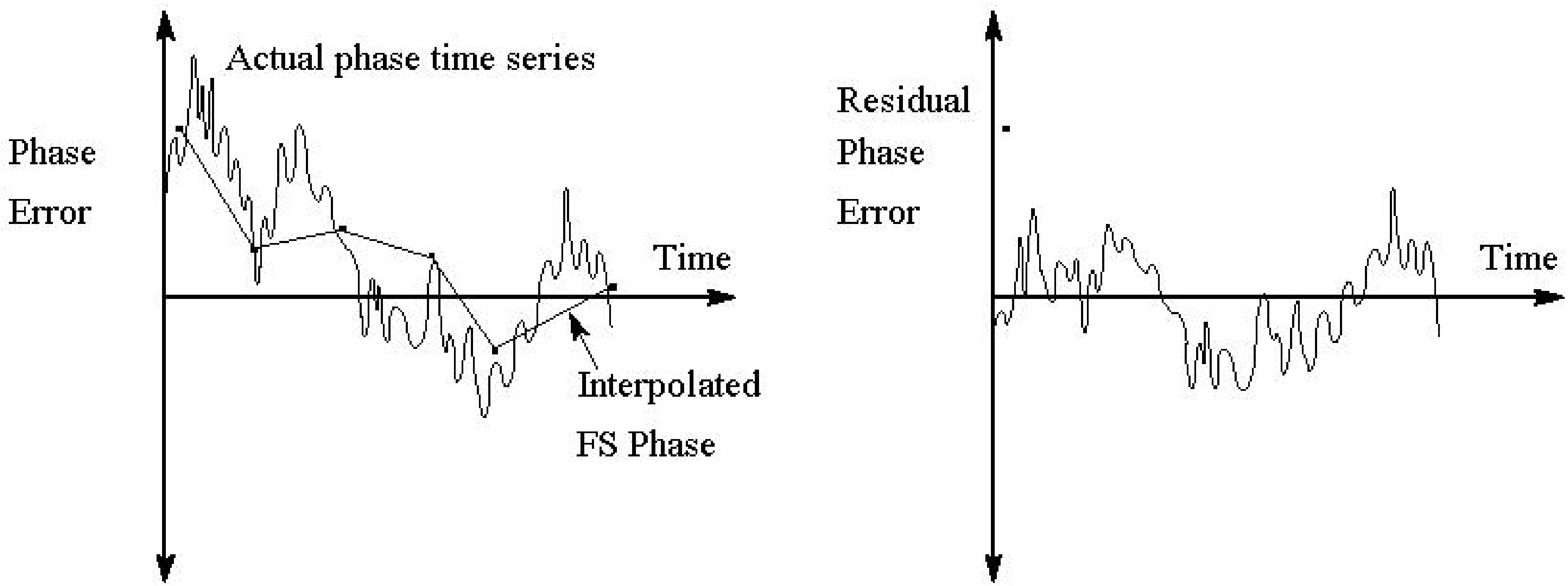


# Some Details on Doing the Simulations

- Choose a model source and configuration which permit good imaging without additional small configurations
- To first order, it doesn't matter what integration time or HA coverage we use, but it is the same for N=50 and N=64
- HOWEVER, long tracks will tend to fill the (u,v) plane, leading to redundancy, reducing the advantage of N=64
- Check sims with error-free case - do we have good (u,v) coverage?
- There are many ways to measure image quality. Choose some.
- Some targeted observations of specific objects will have a scientific observable, a number derived from the image. As image errors average down, such an approach will reduce the contrast between N=50 and N=64.
- To more easily understand the results, look at "slices" through the multi-dimensional phase space.

# Some Details on Simulating Phase Errors

- We are only interested in "residual phase errors" - ie phase errors after calibration.
- Phase calibration: how?  Fast switching:




- Residual phase errors are not Gaussian, but IF the "phase flop time" is smaller than the (u,v) cell crossing time, we can approximate the phase errors as Gaussian, plus a small decorrelation.

# Some Details on Simulating Phase Errors

- We are only interested in “residual phase errors” - ie phase errors after calibration.
- Phase calibration: how?  Fast switching + WVR
- I assert: we need a better model for residual phaser errors after WVR.  Gaussian noise with 1~s time scales on top of slow drifts on the time scale of fast switching cycles
- Start by simulating Gaussian residual phase errors.  Residual phase errors will scale with native phase conditions.

# Regimes to Look For in Phase Self-Cal Simulations:

- Low S: thermal noise limited; there will be a minimum solution time below which we cannot correct for phase errors, lower S means larger minimum solution interval.
  N=64 will have shorter solution intervals, we can track atmospheric changes faster.
  BUT - will it matter in the final images?
- However, with higher SNR in the N=64 case, we will NEED higher SNR on the solution intervals, pushing us to longer solution intervals.

# We Need a Better Theory for Combination of Errors!

sigma^2 = sigma_noise ^ 2 + sigma_phase ^ 2 +....

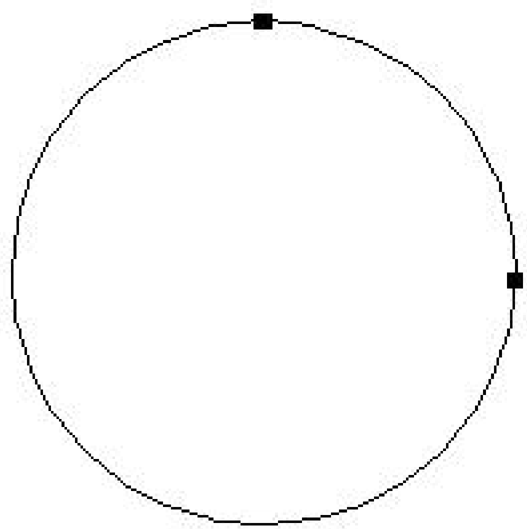
However, this is insufficient as some sigmas will be a function of position. Decorrelation will produce on-source errors, but variable phases will also scatter flux off-source.

# Pointing Error Self Calibration

- Algorithmically much more difficult than phase s.c.
- Built on the conceptual "W-Projection" work of Tim Cornwell
- Multiplication by VP in Image Plane is the same as Convolution by ant. Illumination Pattern in Fourier
- This insight permits imaging w known pointing errors
- P.E. Simulations (w/o self-cal) were pioneered in SDE & CASA will soon contain PE Sim, PE Self-Cal

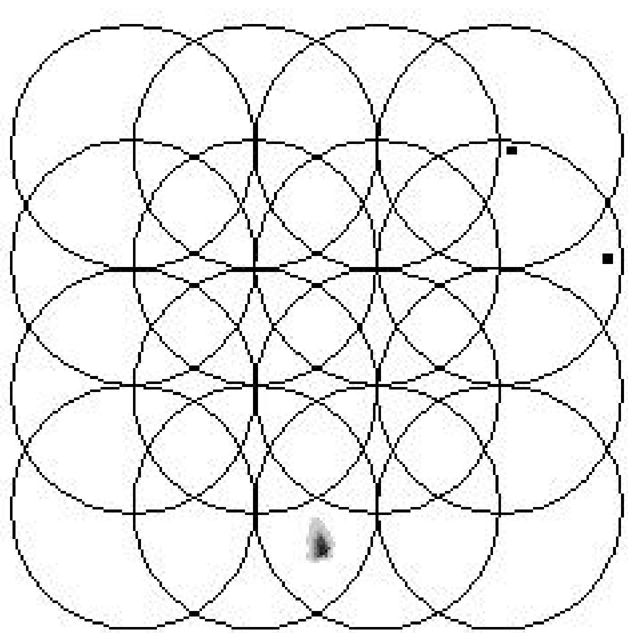
# The Next CASA Release SHOULD Have Full Provisional PE Functionality

- PE Technology is still in its infancy
- Sanjay Bhatnagar is only working on single field
- PE Self-Cal works best when there are multiple (at least 2) bright sources “strategically” located in the beam



# Need to Develop Pointing Self-Cal Intuition, Rules

- We can probably use somewhat extended sources
- The sources need to fill the field - we can use time interpolation as we scan the region repeatedly
- Filter out "noise" in the PE time series



# Misc. Note:
# We also need to worry about simulating Voltage Pattern Errors

- The image errors due to Pointing scale like frequency
- The image errors due to Surface errors scale approximately like freq^2
- IF we have just a few bright sources messing us up, we can solve for ant-dependent amplitude/phase gains on each.
- IF the field is filled with sources, if they are large (ie, cannot be represented by a single complex gain) then we need a different strategy.

I am very happy with my 18 years of service to the ALMA project, and I am excited that it is finally becoming a reality.

But I am even happier and more excited about the Kalimba work I am doing, and the new kinds of catalogs I am creating.

Kalimba CDs are available for the discount rate of 10 Euros.

I have shipped kalimbas to 38 countries, including yours!