

# Tutorial: Forward Modeling & CTEMsoft

Marc De Graef

Carnegie Mellon University

FUNDING: AFOSR MURI FA9550-12-1-0458

# Acknowledgments

- Patrick Callahan, Saransh Singh, Amy Wang, Mike Chapman
- Megna Shaw, Mike Jackson

# CMU 3D Summerschool

- July 8-10, 2015
- <http://materials.cmu.edu/news/summerworkshop.html>
- Speakers: Paul Salvador, Tony Rollett, Greg Rohrer, Dave Rowenhorst, Liz Holm, Bob Suter, Marc De Graef, Mike Groeber, Mike Jackson, Shawn Litster
- Day 3: Dream.3D data analysis OR plugin development

# Outline

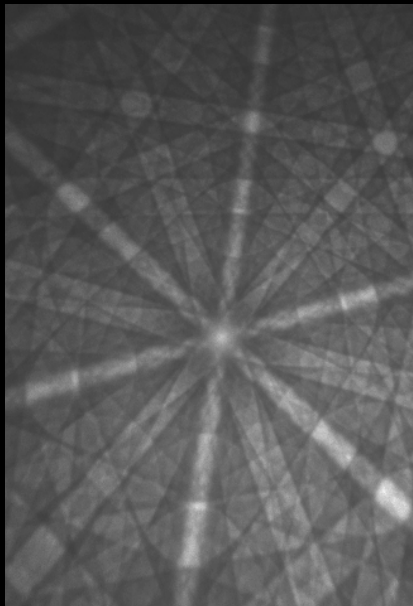
- What is forward modeling? and why do we need it?
- CTEMsoft overview
- Application to EBSD
  - simulation pipeline
  - examples
- Dictionary-based indexing
- Future directions

# Motivation

In materials characterization, we employ many different modalities to explore the structure and chemistry of a material.

TEM, LTEM, CBED, LACBED, EDS, EELS, ALCHEMI, SE, BSE, EBSD, ECP, ECCI, ...

## EBSD



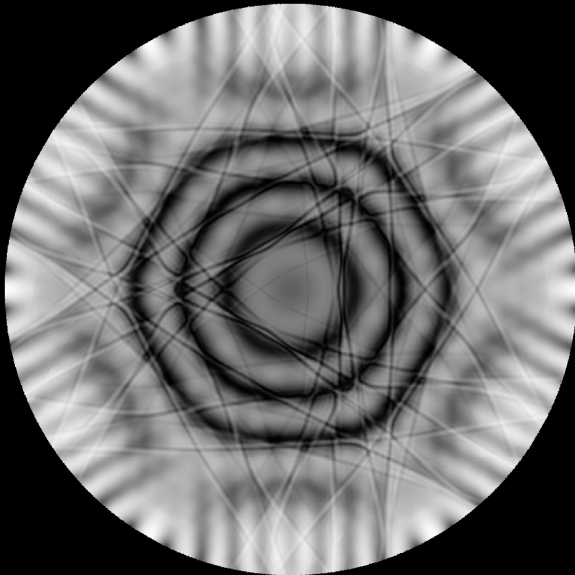
- back-scattered electrons carry information about
  - lattice orientation
  - surface topography
  - chemistry (including ordering)
- standard EBSD implementations only use pattern geometry and ignore intensity distribution; thus, we throw away most of the data
- full interpretation requires a physics-based understanding of how the pattern is formed

# Motivation

In materials characterization, we employ many different modalities to explore the structure and chemistry of a material.

TEM, LTEM, CBED, LACBED, EDS, EELS, ALCHEMI, SE, BSE, EBSD, ECP, ECCI, ...

## CBED



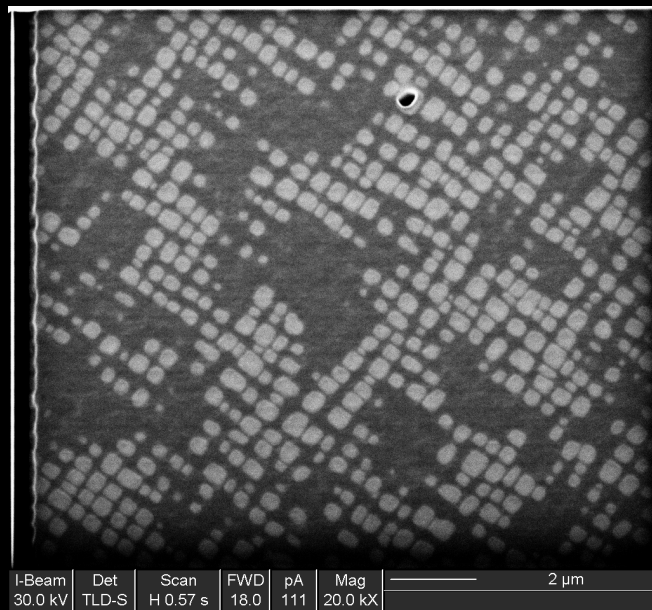
- dynamically scattered electrons carry information about
  - lattice orientation
  - lattice symmetry (full space group)
  - chemistry (including ordering)
  - atomic bonds & lattice potential
  - lattice strain
- CBED patterns are usually interpreted visually
- full interpretation requires a physics-based understanding of how the pattern is formed

# Motivation

In materials characterization, we employ many different modalities to explore the structure and chemistry of a material.

TEM, LTEM, CBED, LACBED, EDS, EELS, ALCHEMI, SE, BSE, EBSD, ECP, ECCI, ...

## SE/BSE/ISE

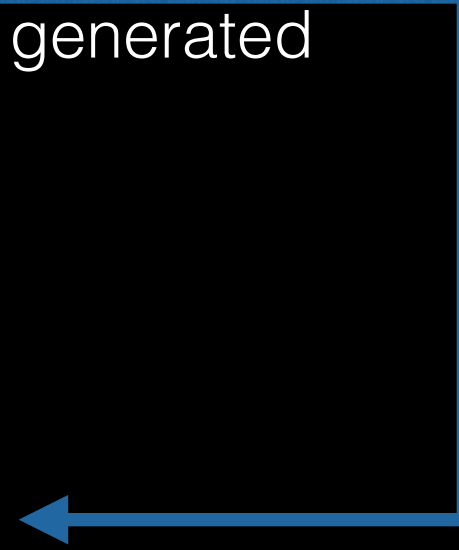


- inelastically scattered electrons carry information about
  - phase
  - chemistry
  - orientation
  - topography
- SEM-based images are usually interpreted visually
- full interpretation requires a physics-based understanding of how the pattern is formed

# Motivation

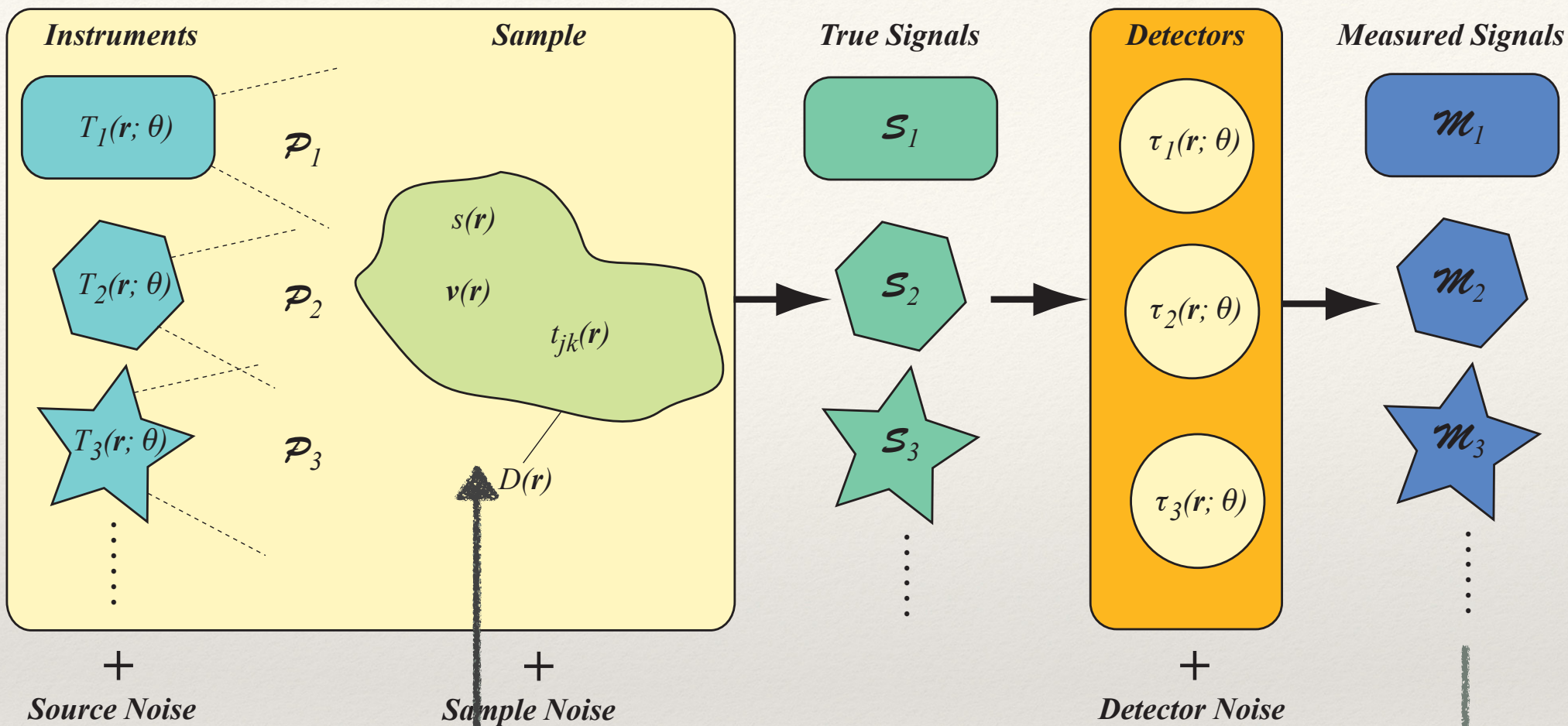
full interpretation requires a physics-based understanding  
of how the signal is generated

- quantitative analysis
- error/uncertainty estimation
- validation





# GENERAL DESCRIPTION OF MATERIALS CHARACTERIZATION



Generalized forward projector (physics-based)

$$\mathcal{M}_i = \mathcal{P}_i [D(\mathbf{r}), s(\mathbf{r}), \nu(\mathbf{r}), t_{jk}(\mathbf{r}), \dots; T_i(\mathbf{r}, \theta); \tau_i(\mathbf{r}, \theta'); \text{noise terms}]$$

unknown

can be modeled

---

# Ingredients of a “good” forward model

---

- ❖ accurate model for incident probe (electron, photon, ...)
- ❖ accurate physics of probe-sample interactions
- ❖ accurate models for detector behavior
- ❖ proper accounting for noise and stochastic nature of scattering processes
- ❖ flexibility to independently modify each of the above items
- ❖ efficient implementation (fast and portable)

# Forward modeling and DREAM.3D

- Given a serial sectioning EBSD data set, and a reconstructed microstructure, how good is this reconstruction?
  - Create 3D microstructure with a given statistics extracted from a serial sectioning experiment (e.g., a stack of \*.ang files)
  - Define a serial sectioning geometry, as well as an EBSD detector geometry and sampling grid
  - Compute (via forward modeling) an EBSD dataset for this serial sectioning model
  - Index the EBSD dataset (either via OEM or dictionary approaches)
  - Reconstruct the microstructure using DREAM.3D
  - Compare with the starting microstructure.

# What is CTEMsoft?

- mid 90s to 2001: assorted f77 routines for electron scattering simulations
- 2002-2003: complete f90 rewrite for TEM book; partial v1.0 made available via book web site
- 2004-2012: expansion of library, addition of Monte Carlo and SEM-based modalities; restructuring in context of forward modeling
- 2013: sabbatical at OSU, complete rewrite of everything
- June 2014: v2.0 made available via github
- July 2014 - now: complete rewrite (again), addition of GPU, HDF format, f2003, interface to C++ for inclusion with DREAM.3D, etc...
- Fall 2015: v3.0 source code available, renamed to **EMsoft**

What follows will be Mac-centric.

It's not that we don't like Windows...

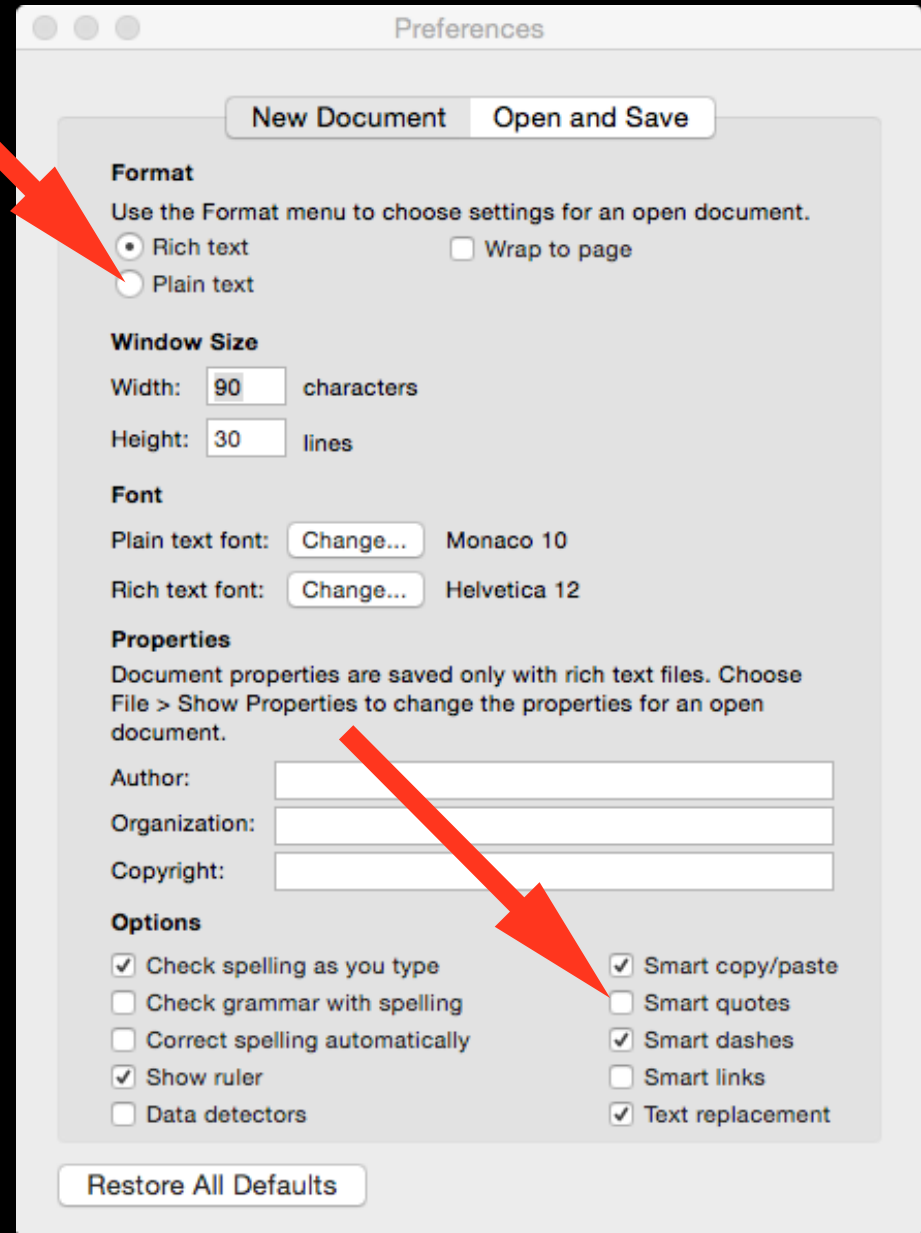
It's just that we don't have any computers that run Windows...

If you try any of the following on a Windows computer, please let us know how it went, and also what you had to do to make it all work...

From here on, everything will be valid only for v2.0; v3.0 will have major changes

# text editing comment

- If you know vi, vim, emacs, or any other text editing program, just use them as before
- If you use the TextEdit app, you must use Plain Text and turn off the Smart Quotes option in the Preferences Panel...



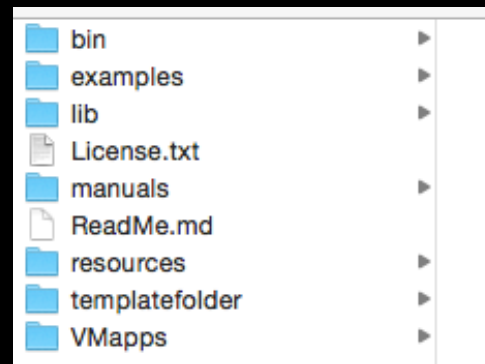
# Obtaining the software

- Source Code [CMake, Xcode, gcc, gfortran, blas, lapack]
  - <http://www.github.com/marcdegraef/CTEMsoft>
- Executables (Mac OS X 10.8 and more recent)
  - <http://materials.cmu.edu/degraef/CTEMsoft-2.0-OSX.zip>
- Windows executables currently unavailable; could use some help with this ...

Install Demo

# Setting up the environment variables

- We'll assume that you are using the C-shell; other shells require similar environment variables...
- Say you install the package in `/Users/me/CTEMsoft`; the folder will have the following structure:



- in your `~/.cshrc` start up file, add the following lines:  

```
setenv CTEMsoft2013templates /Users/me/CTEMsoft/templatefolder  
setenv CTEMsoft2013resources /Users/me/CTEMsoft/resources
```
- and restart your shell (`exec csh`)  

```
set path = ($path /Users/me/CTEMsoft/bin)
```



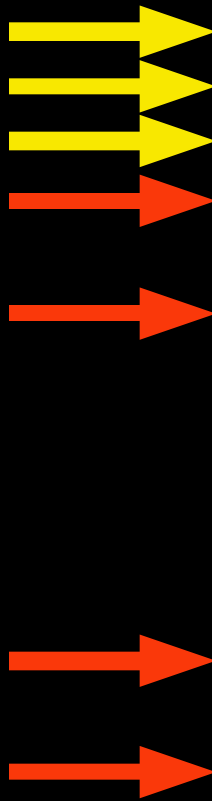
# Create your workfolder

- It's a good idea to keep all your CTEMsoft-related files together, so let's create a workfolder;
  - Example: in /Users/me, execute the commands

```
mkdir playarea  
cd playarea
```

# Available programs (23)

Program	Short description	Manual?
CTEMLacbed	large angle CBED pattern computation (TEM)	✓
CTEMmbcbcd	regular CBED pattern (TEM)	✓
CTEMSRdefect	(S)TEM systematic row defect images (TEM)	✓
CTEMZAdefect	(S)TEM zone axis defect images (TEM)	✓
CTEMECP	electron channeling pattern (SEM)	✓
CTEMECCI	ECCI defect images (SEM)	✓
CTEMKossel	electron Kossel patterns (SEM)	✓
CTEMMC	Monte Carlo trajectory simulations (SEM)	✓
CTEMEBSDmaster	EBSD master pattern simulation (SEM)	✓
CTEMEBSD	EBSD pattern simulation (SEM)	✓
CTEMmkxtal	make a crystal structure input file	✓
CTEMlistSG	list the equivalent positions of any space group	
CTEMqg	list Fourier coefficient information	
CTEMfamily	draw a family of planes (stereographic)	
CTEMstar	list the star of any reciprocal lattice point	
CTEMorbit	list the orbit of a point	
CTEMZAgeom	print zone axis geometry and symmetry information	
CTEMlatgeom	perform some simple lattice geometry calculations	
CTEMstereo	basic stereographic projection tool	
CTEMorient	stereographic projection for orientation relation	
CTEMxtalinfo	makes postscript file with lots of information	
CTEMzap	create postscript file with zone axis patterns	
CTEMdrawcell	draw a unit cell (very primitive)	



# .xtal files

- ALL CTEMsoft programs need a crystal structure input file, which can be created by the CTEMmkxtal program.
- Requirements:
  - lattice parameters
  - space group number (and origin setting)
  - asymmetric unit, including site occupancies and Debye-Waller factors

CTEMmkxtal demo

# simple programs

- bottom half of program list ...

CTEMstereo demo

CTEMqg demo

CTEMxtalinfo demo

- for programs that generate a postscript file, simply type “open filename.eps” and the file will be converted to pdf and opened in Preview ...

# CTEMsoft input style

- programs use one of two input methods
  - interactive (bottom half of table on previous slide)
  - namelist file (top half)
- Future version will replace these with a Python GUI and HDF5 input files; Dream.3D plugins/filters are also being planned...
- V2.0 requires basic UNIX command line knowledge

# CTEMstereo

- produces a basic stereographic projection of your crystal structure along a user-definable direction; allows for direct or reciprocal space projections.
- output: a PostScript file; can be displayed in Preview by typing ``open file.ps'' on the command line
- very simple program; useful for TEM navigation and pattern indexing

Demo of CTEMstereo

# CTEM<sub>qg</sub>

- displays potential Fourier coefficients for selected lattice planes in a number of different forms, including  $1/q$  which appears in the dynamical scattering equations:

$$\frac{d\psi_{\mathbf{g}}}{dz} - 2\pi i s_{\mathbf{g}} \psi_{\mathbf{g}} = i\pi \sum_{\mathbf{g}'} \frac{e^{i\theta_{\mathbf{g}-\mathbf{g}'}}}{q_{\mathbf{g}-\mathbf{g}'}} \psi_{\mathbf{g}'}$$

Demo of CTEM<sub>qg</sub>

# CTEMxtalinfo

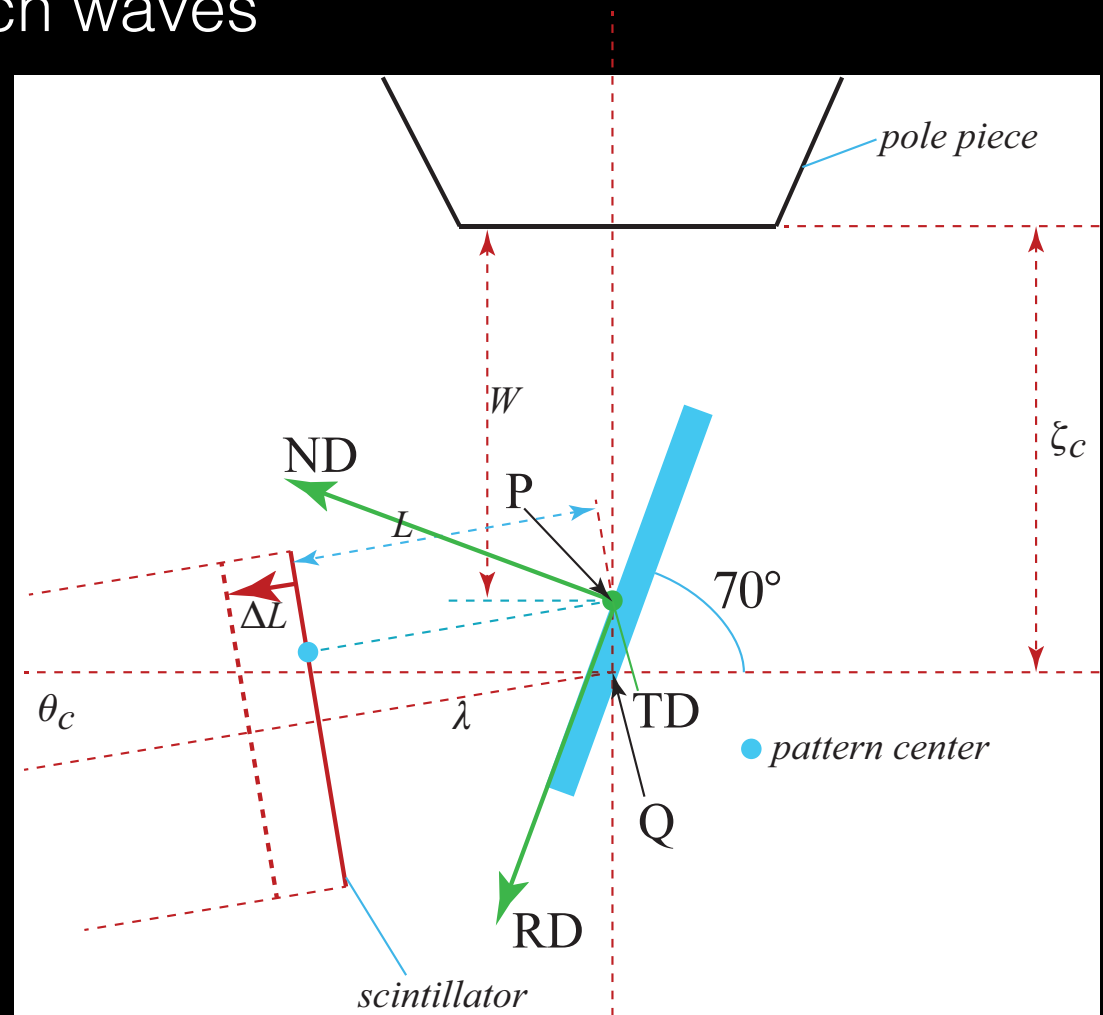
- generates a PostScript file with
  - unit cell geometry, including various tensors
  - scattering information for allowed reflection families
  - stereographic projections
  - kinematical zone axis diffraction patterns
- useful when starting TEM work on a new crystal structure

Demo of CTEMxtalinfo



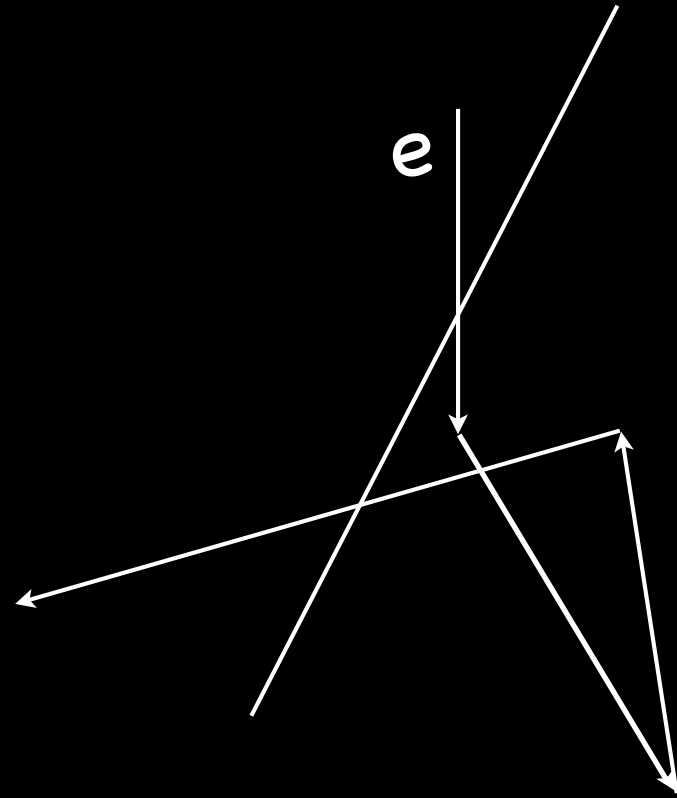
# EBSD simulation pipeline

- Simulation consists of three steps:
  - stochastic inelastic scattering: Monte Carlo
  - dynamical scattering: Bloch waves
  - detector geometry



# Step 1: Monte Carlo simulations

- Elementary step
  - travel some distance
  - loose some energy
  - change direction
- Repeat until  $e$  leaves the sample, or energy falls below threshold value



CTEMMC program

- Distance traveled between scattering events is determined by the mean free path, which is described by a probability distribution
  - we can sample this distribution so that the resulting distribution of path lengths has the same mean value
- The angular distribution after scattering follows from knowledge of differential cross sections for different scattering processes (Rutherford, core excitation, etc)

# Continuous Slowing Down Approximation (CSDA)

- Simple model for energy loss: continuous Bethe stopping power

- estimate of energy loss per unit path length

$$S_B = 7.85 \times 10^3 \frac{Z\rho}{AE} \ln \left( \frac{1.166E}{J} \right) \text{ [eV/nm]}$$

$$J/Z = 0.015 \quad Z \leq 13;$$

$$J/Z = 9.76 + 58.5^{-0.19} \quad Z > 13$$

Several modifications of this relation are in use. This approach ignores core excitations, which correspond to a large drop of energy in a single event.

# CTEMMC inputs

```
&MCdata
! name of the crystal structure file
xtalname = 'Al.xtal'
! sample tilt angle from horizontal [degrees]
sig = 70.0
! sample tilt angle around RD axis [degrees]
omega = 0.0
! number of pixels along x-direction of square projection [odd number!]
numsx = 751
! seed for Monte Carlo random number generator [should be a prime number]
primeseed = 932117
! number of incident electrons per thread
num_el = 12500000
! number of OpenMP threads
nthreads = 4
! incident beam energy [keV]
EkeV = 30.D0
! minimum energy to consider [keV]
Ehistmin = 25.D0
! energy bin size [keV]
Ebinsize = 1.0D0
! maximum depth to consider for exit depth statistics [nm]
depthmax = 100.D0
! depth step size [nm]
depthstep = 1.0D0
! output data file name
dataname = '/Users/me/playarea/MC_Al.data'
```

Takes about 25 minutes  
on 2011 Macbook Pro,  
4 threads

V3.0: GPU version  
300 times faster ...

# IDL Display Program

- data visualization is important.
- dedicated IDL (Interactive Data Language) scripts are available for analyzing the CTEMsoft data files.
- No IDL license needed; pre-compiled into “virtual machine apps” (VMapps); should work without installing anything, in principle ... **BUT idl is not officially supported on OS X 10.10 (Yosemite)**...
- Check the VMapps folder in the distribution for the ***EBSDDisplay*** app; double click (make sure XQuartz is available);

Demo visualization of CTEM MC output

# Step 2: EBSD master pattern

Structure factor-like

Bloch wave  
dynamical simulation

$$\begin{aligned}
 \mathcal{P}(\mathbf{k}_0) &= \sum_{\mathbf{g}} \sum_{\mathbf{h}} S_{\mathbf{g}\mathbf{h}} L_{\mathbf{g}\mathbf{h}}, \\
 S_{\mathbf{g}\mathbf{h}} &\equiv \sum_n \sum_{i \in \mathcal{S}_n} Z_n^2 e^{-M_{\mathbf{h}-\mathbf{g}}^{(n)}} e^{2\pi i(\mathbf{h}-\mathbf{g}) \cdot \mathbf{r}_i}, \\
 L_{\mathbf{g}\mathbf{h}} &\equiv \sum_j \sum_k C_{\mathbf{g}}^{(j)*} \alpha^{(j)*} \mathcal{I}_{jk} \alpha^{(k)} C_{\mathbf{h}}^{(k)}.
 \end{aligned}$$

$$\mathcal{I}_{jk} = \frac{1}{z_0} \int_0^{z_0} e^{-2\pi(\alpha_{jk} + i\beta_{jk})} dz$$

$$\mathcal{I}_{jk} = \frac{1}{z_0(E)} \int_0^{z_0(E)} \lambda(E, z) e^{-2\pi(\alpha_{jk} + i\beta_{jk})} dz$$

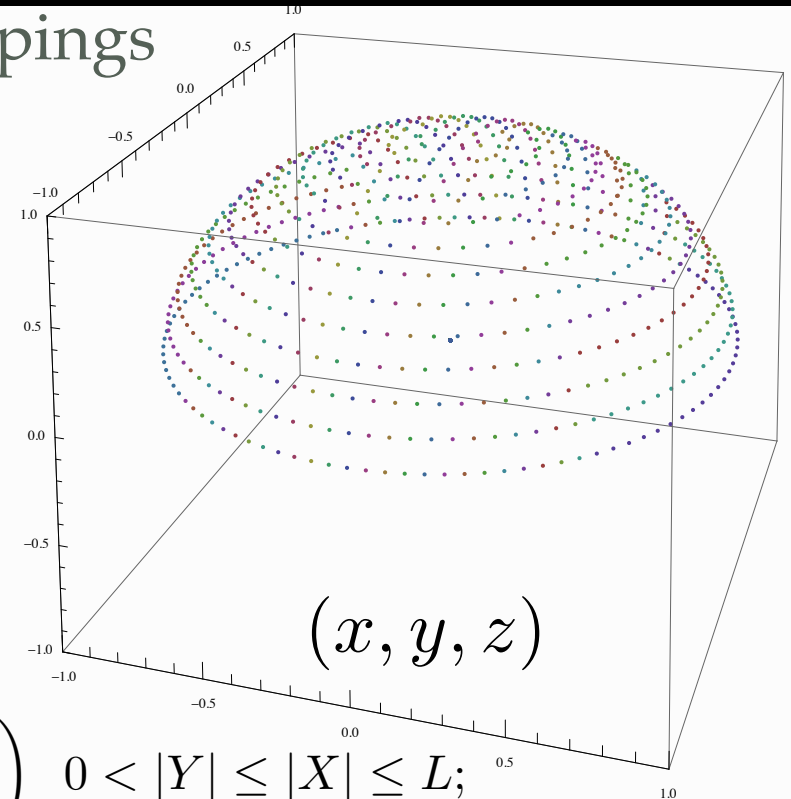
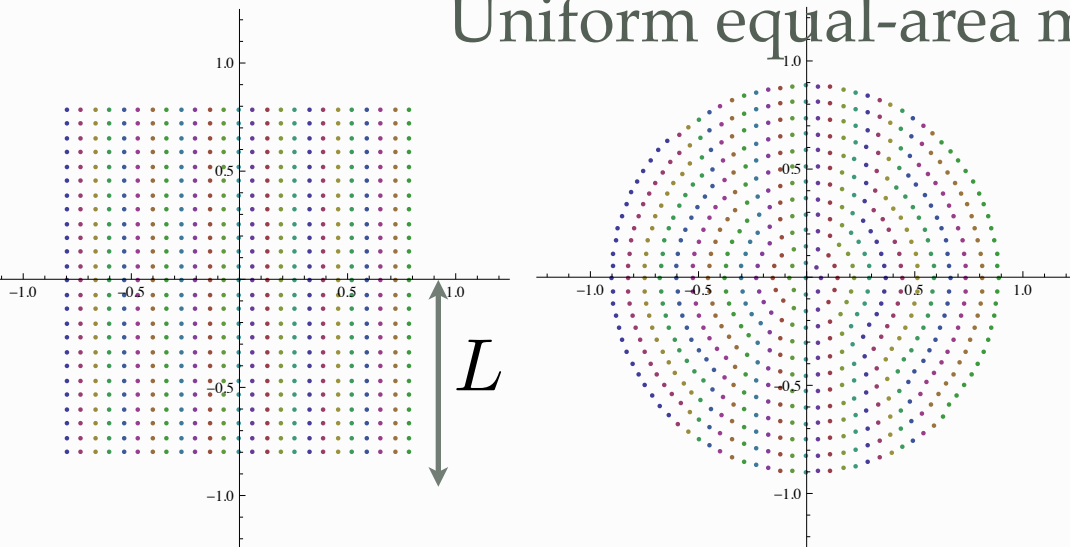
from MC

$$I_{\text{BSE}}(i, j) = \eta \sum_{\kappa=1}^{n_E} \mathcal{E}(\kappa, i, j) \mathcal{M}(\kappa, X(x_s, y_s, \tilde{q}), Y(x_s, y_s, \tilde{q}))$$

# EBSD master pattern

- for computational efficiency, master patterns are stored in modified Lambert projection form

## Uniform equal-area mappings



$$(X, Y)$$

$$(x, y, z) = \left( c(X) \cos \frac{Y\pi}{4X}, c(X) \sin \frac{Y\pi}{4X}, 1 - \frac{2X^2}{\pi} \right) \quad 0 < |Y| \leq |X| \leq L;$$

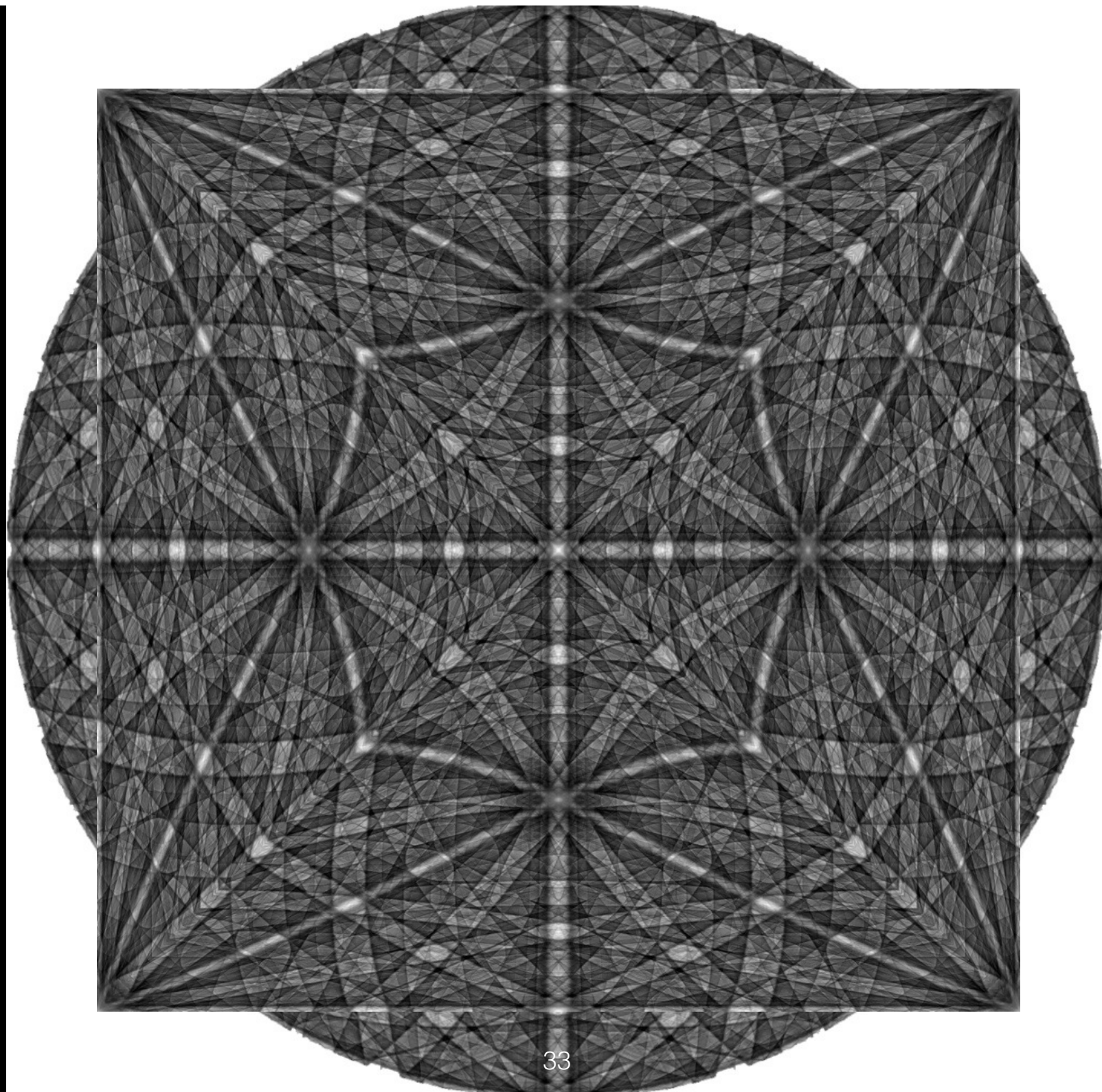
$$(x, y, z) = \left( c(Y) \sin \frac{X\pi}{4Y}, c(Y) \cos \frac{X\pi}{4Y}, 1 - \frac{2Y^2}{\pi} \right) \quad 0 < |X| \leq |Y| \leq L,$$

$$L = \frac{\pi}{\sqrt{2}}$$

where

$$c(p) = \frac{2p}{\pi} \sqrt{\pi - p^2}.$$





Ni  
30 kV

# CTEMEBSDmaster inputs

&EBSDmastervars

! smallest d-spacing to take into account [nm]

dmin = 0.05

! number of pixels along x-direction of the square master pattern ( $2 \cdot \text{npx} + 1 = \text{total number}$ )

npx = 500

! name of output file (to be read by IDL visualization routine)

outname = '/Users/me/playarea/Al\_master.data'

! name of the energy statistics file produced by CTEMMC program [note that

! this file will also contain parameters such as the microscope accelerating voltage]

energyfile = '/Users/me/playarea/Al\_MC.data'

/

Takes about 30 minutes  
on 2011 Macbook Pro,  
1 thread

Note that the v2.0 CTEMEBSDmaster program is a serial program; therefore, it can take a very long time to complete a single energy-set of master patterns...

v3.0 runs on a GPU, using a scattering matrix algorithm instead of the Bloch wave approach

EBSDDisplay demo

# Step 3: Detector Geometry

## CTEMEBSD program

- Can be executed from command line, with .nml input file; in this mode one would compute either a single hi-res pattern, or a large dictionary of patterns based on an input Euler angle list.
- Can also be executed from within the IDL visualization program, to provide a more direct interaction; in this mode one can study the effect of detector parameter changes, and compute single patterns or short lists of patterns.
- This program is very fast since the pattern computation only requires bilinear interpolation of the master patterns.

```
&EBSDdata
! distance between scintillator and illumination point [microns]
L = 15250.0
! tilt angle of the camera (positive below horizontal, [degrees])
thetac = 0.0
! CCD pixel size on the scintillator surface [microns]
delta = 49.375
! number of CCD pixels along x and y
numsx = 640
numsy = 480
! pattern center coordinates in units of pixels
xpc = 3.57
ypc = 113.45
! name of angle file (euler angles or quaternions)
anglefile = 'testeuler.txt'
! 'tsl' or 'hkl' Euler angle convention parameter
eulerconvention = 'tsl'
! name of EBSD master output file
FZfile = '/Users/me/playarea/AI_master.data'
! name of Monte Carlo output file
energyfile = '/Users/me/playarea/AI_MC.data'
! name of output file
datafile = '/Users/me/playarea/EBSDtest.data'
! incident beam current [nA]
beamcurrent = 150.0
! beam dwell time [micro s]
dwelltime = 100.0
! binning mode (1, 2, 4, or 8)
binning = 1
```

EBSDDisplay demo

# EBSD Summary

- EBSD pattern simulation (dynamical) is broken down into 3 steps:
  - Monte Carlo: energy, depth, and directional distributions
  - Master pattern: dynamical simulation for all possible directions
  - Pattern computation: application of detector parameters
- MC and MP are lengthy computations, final step is interactive
- v3.0: most of this will run on GPU

# CTEMsoft future

- Fall 2015: v3.0 **EMsoft** release; OpenMP/GPU, HDF5
- Integration with Dream.3D [EBSD portion]
  - filters to create EMsoft input files
  - filters to run the individual programs
  - filters to create simulated EBSD data sets for a synthetic microstructure.

# New Indexing and Segmentation Approach

Collaboration with Al Hero and his students at UMich,  
Jeff Simmons, Mike Uchic, Mike Groeber,  
Megna Shah, Mike Jackson

- \* Method is based on a dictionary of pre-computed EBSD patterns.
- \* use dynamical simulation approach to generate all the patterns
- \* requires uniform sampling of orientation space  $SO(3)$ , at least the part of it that corresponds to a crystallographic fundamental zone
- \* FZs are simplest in Rodrigues space (planar bounds)

# new vs. old EBSD indexing

- \* Ni test sample, 10 runs on same ROI [Stuart Wright, EDAX]
- \* Different acquisition conditions, constant exposure
- \* Indexed with standard TSL approach as well as with dictionary method based on forward model...
  - \* 333,227 patterns in dictionary, 60x60 pixels, 20kV
  - \* cubochoric sampling of Rodrigues cubic fundamental zone



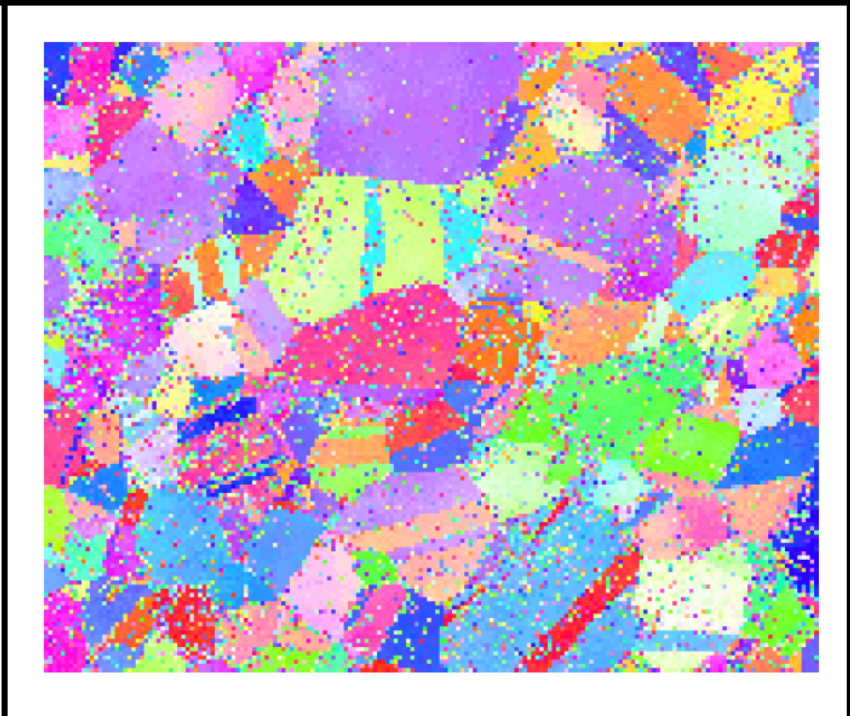
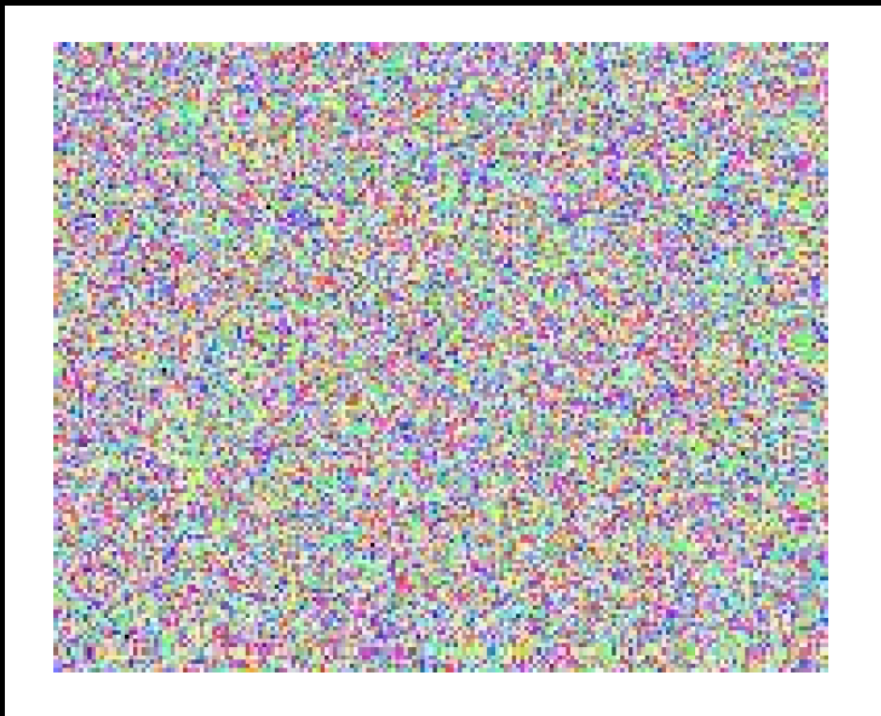
<b>Scan</b>	<b>Gain</b>	<b>Gain [dB]</b>	<b>Camera Exposure msec]</b>	<b>Exposure Level</b>	<b>Frame Rate</b>	<b>Indexing Speed</b>	<b>CI Indexing Success Rate</b>
<b>1</b>	512	18.01759531	0.94	90%	1073	960	99.1
<b>2</b>	768	27.02639296	0.32	90%	1073	960	90.6
<b>3</b>	640	22.52199413	0.55	90%	1073	960	98.4
<b>4</b>	896	31.53079179	0.17	90%	1073	960	19.4
<b>5</b>	1023	36	0.1	90%	1073	960	0.1
<b>6</b>	832	29.27859238	0.23	90%	1073	960	60.2
<b>7</b>	800	28.15249267	0.27	90%	1073	960	77.2
<b>8</b>	864	30.40469208	0.2	90%	1073	960	37.3
<b>9</b>	256	9.008797654	2.78	90%	359	359	99.7
<b>10</b>	0	0	8.03	90%	124	124	99.8

**CI Index  
Rate**

**OIM Analysis Result**

**Dictionary Result**

0.1



Questions ?