Moving ROOT Forward

Balancing between stability and innovation

Fons Rademakers
CERN

ACAT'11, 8-Sept, 2011, Brunel Univ., U.K..

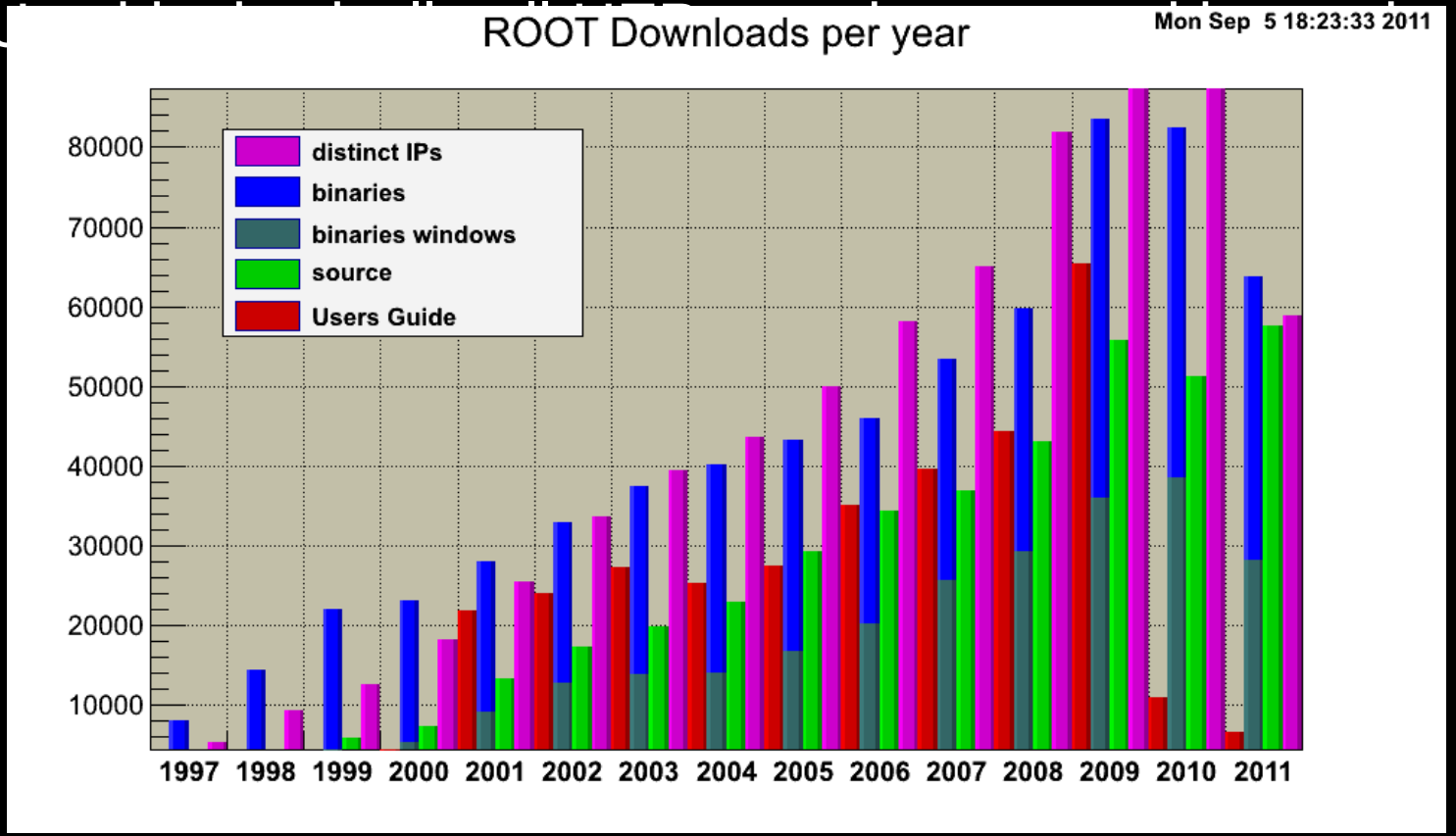
Outline

- Interpreter
- I/O
- Statistical tools
- PROOF
- Pads and Phones
- Browsers and HTML5
- Graphics
- Infrastructure
- Conclusions

Disclaimer: we can't commit to specific release dates for these new features

ROOT Going Strong

- Ever increasing number of users
  - 5600 forum members, 56850 posts, 1300 mailing list members
  - U



New Interpreter

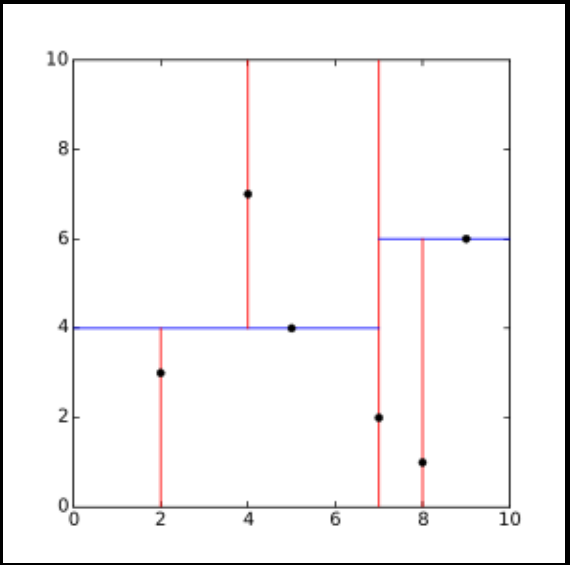
- Replacing good old CINT by new, shining Cling
- Cling is based on LLVM and Clang compiler libraries
  - LLVM/Clang “new” open source compiler suite
  - Default compiler on OSX Lion
- Cling released in July 2011
  - Fully functional C and C++ interpreter (including C++11)
  - Uses Just-In-Time compilation
  - Still a few issues to solve (e.g. reloading of code)
- Integration with ROOT starting now
  - Interfaces via TCling and TClass
  - Use precompiled header files (PCH’s) as dictionary repository (no more huge compiled dictionaries)

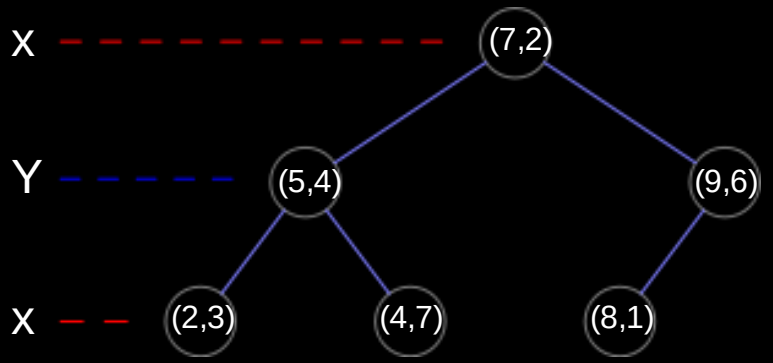
Improvements in I/O

- Parallel Tree merging
- Smaller but important improvements
  - Reimplementation of OptimizeBaskets
  - I/O Customization: Nested Objects
  - Explore changing the on-file byte format to little endian
  - ...
- Support for Amazon S3, CloudFront and Google Storage
- Continuous review of I/O performance in close collaboration with experiment experts

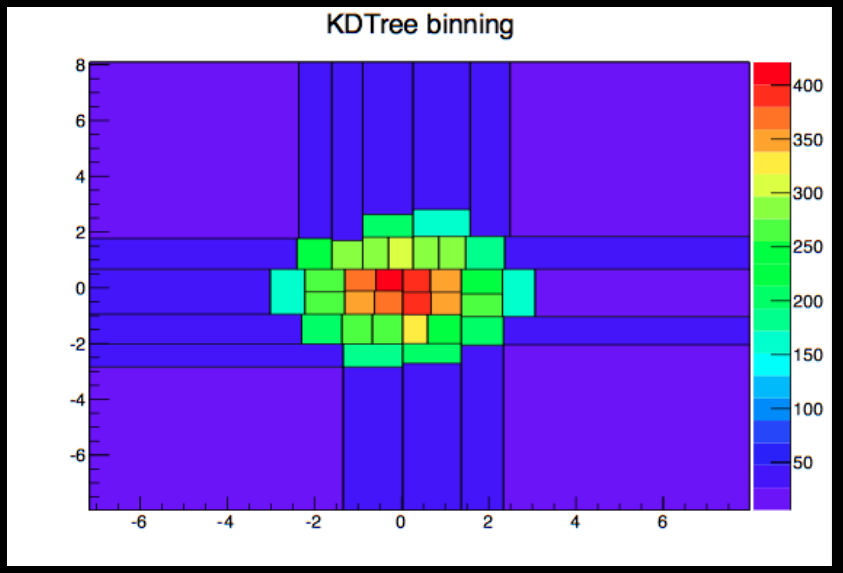
Improvements in Statistical Tools

- kd tree: space partitioning for multi-dimensional data structure (optimized for range or closest point searches)



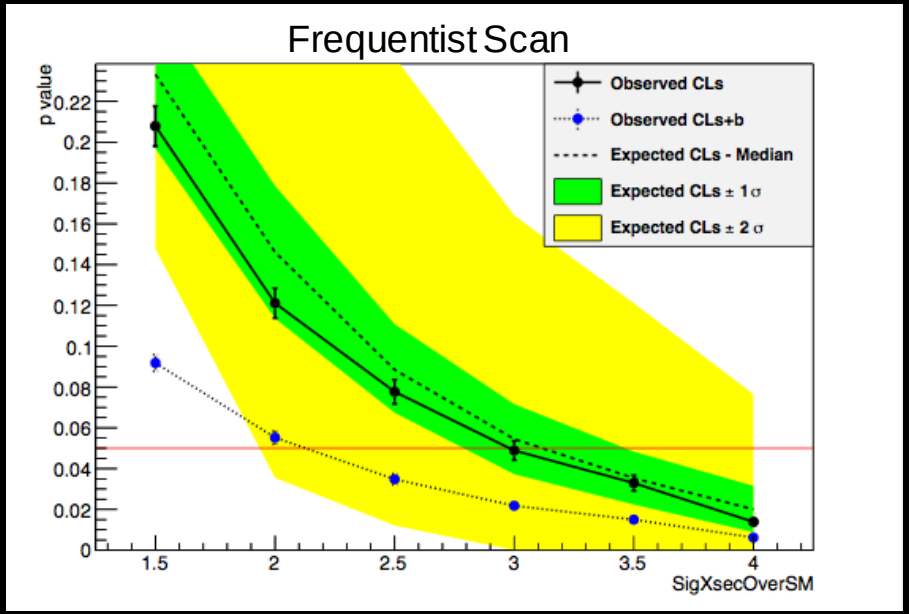


- New class for binning multi-dimensional data using a kd tree
  - Efficient multi-dimensional histogram
  - Can be used for non-parametric density estimation using kernels



New RooStats Features

- Classes for computing limits using procedures endorsed and agreed by the LHC experiments (frequentist CLs limits)
  - Can compute observed limit and expected limits with sigma bands using toy Monte Carlo events
- Being used by several analysis group to report their limit results
- Can be very time consuming, since many toys need to be generated and fitted
  - Working on improving performances
  - More efficient fitting for multi-parameter problems
  - Faster toy generation



Improvements in PROOF

- Improved connection layout, remove single point of failure, remove interference between users
- Result merging optimization (see also parallel tree merging in I/O)
- Packetizer redesign
  - Large number of events of ~same size (analysis)
  - Small number of events of different sizes (reconstruction)
  - Non-homogenous data distribution (local, networked data)
  - Dynamically handle new work, i.e. new files
- Multi-Master improvements
  - Dynamic re-organization of a set of workers in a multi-tier structure
  - MM dataset manager
  - MM packetizer for dynamic load-balancing across sub-masters

Tablets and Smart Phones

- Currently 200 Million iOS devices, 25 Millions iPads
- All of ROOT (except device graphics) ported to iOS (little endian, 32-bit, ARM)
  - To cross compile on OSX the iOS native version: ./configure ios
  - To compile on OSX the iOS simulator version: ./configure iossim
- Android may come later, but not many tablets yet, so lower priority

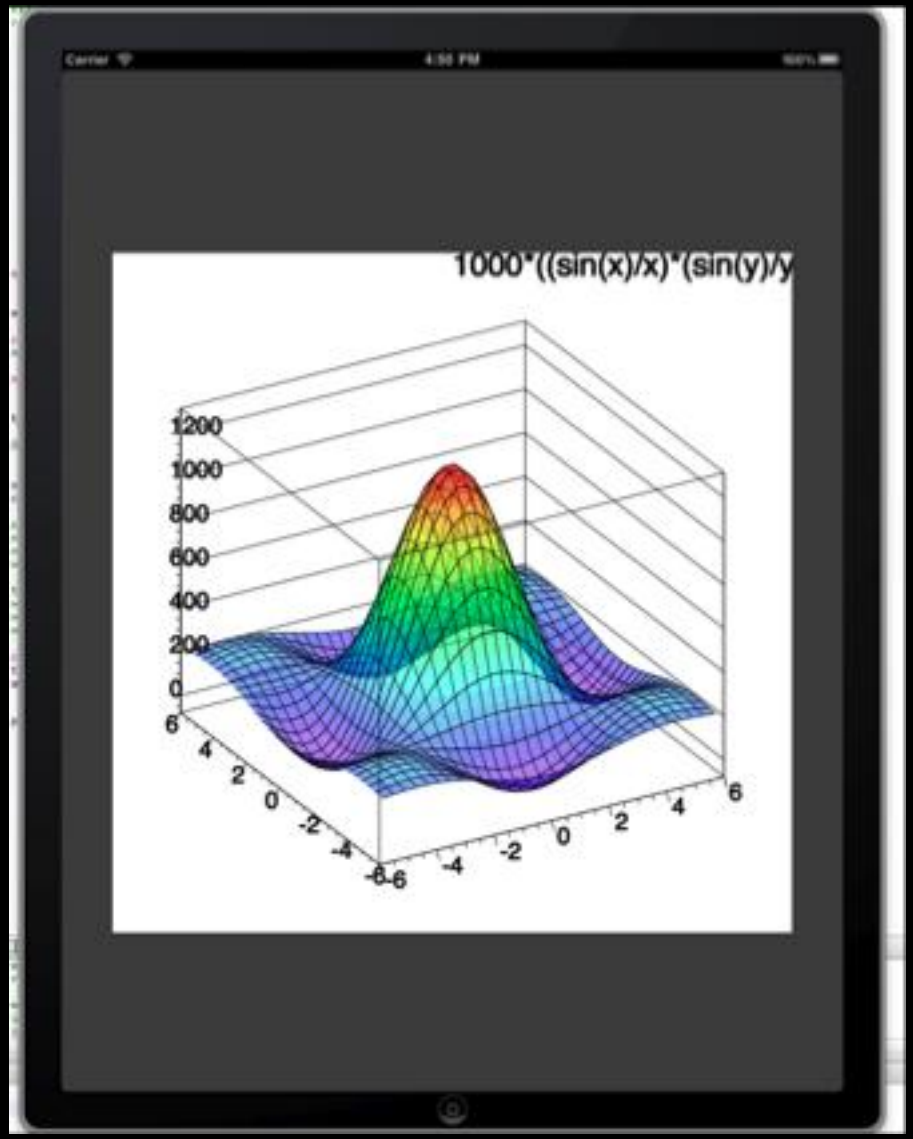
Porting to iOS - iOS is Unix, but...

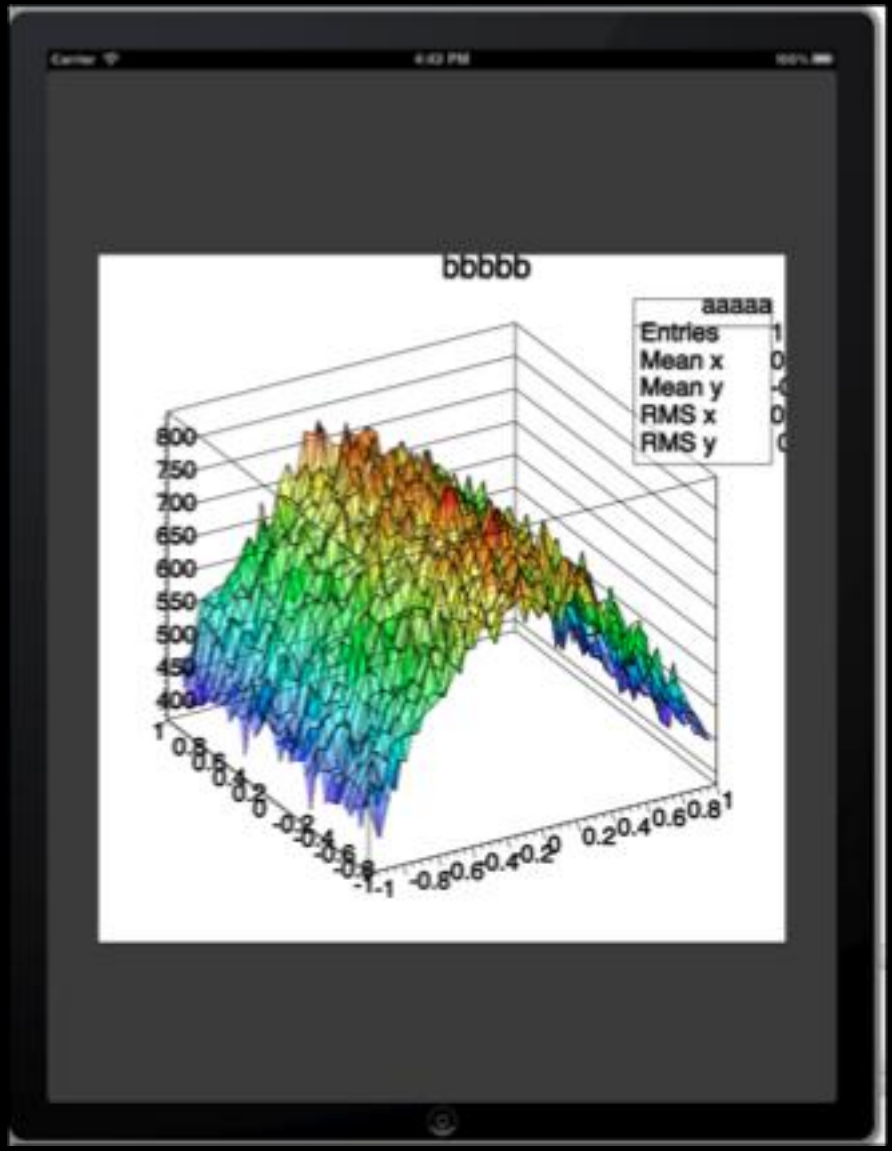
- No X11 (no TVirtualX, no ROOT GUI)
- No user shared libraries, no plugins (only system shared libraries)
  - ROOT builds into a single libROOT.a
- Objective-C/Objective-C++ to create native GUI
- Rich set of libraries and frameworks: CoreGraphics, CoreFoundations, CoreText, CoreAnimation, UIKit, GLES, etc.
- Specific touch GUI, different from Windows/X11/OSX
- Fingers instead of mouse, touches instead of clicks, no cursor

Implementing ROOT Graphics on iOS

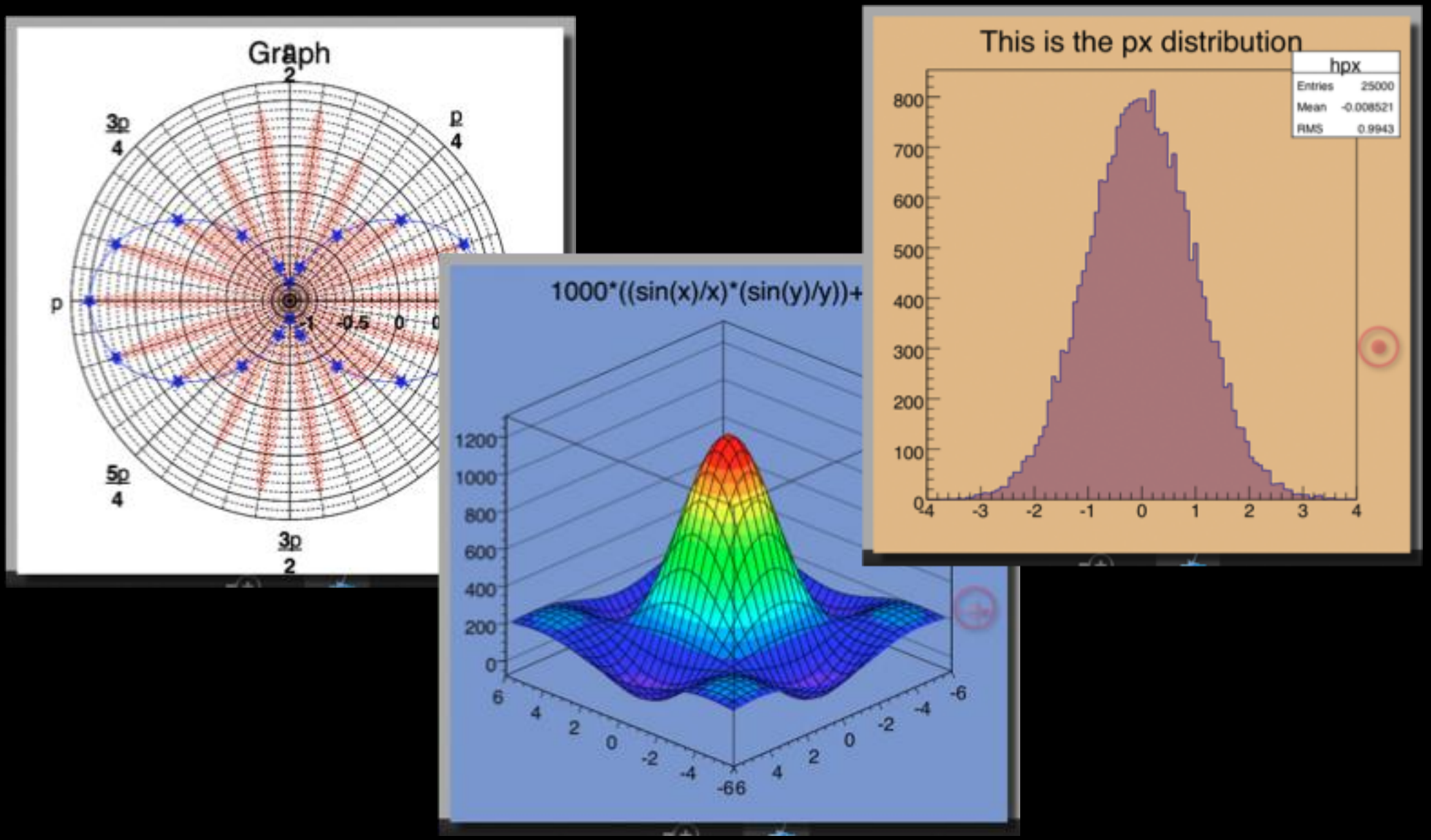
- Ported ROOT's non-GUI graphics on iOS
  - TVirtualPad interface in ROOT::iOS::Pad using Quartz 2D library
- Writing ROOT-based native applications for iOS
  - Demo of ROOT's graphics
  - Browser for ROOT files
  - Components to support users who want to develop for iOS using ROOT
- Port ROOT's OpenGL code to GLES
- Use HPC (head coupled perspective) technology for event display (Eve)
  - Uses front facing camera
  - Glasses free 3D display

First Version (Middle of June)

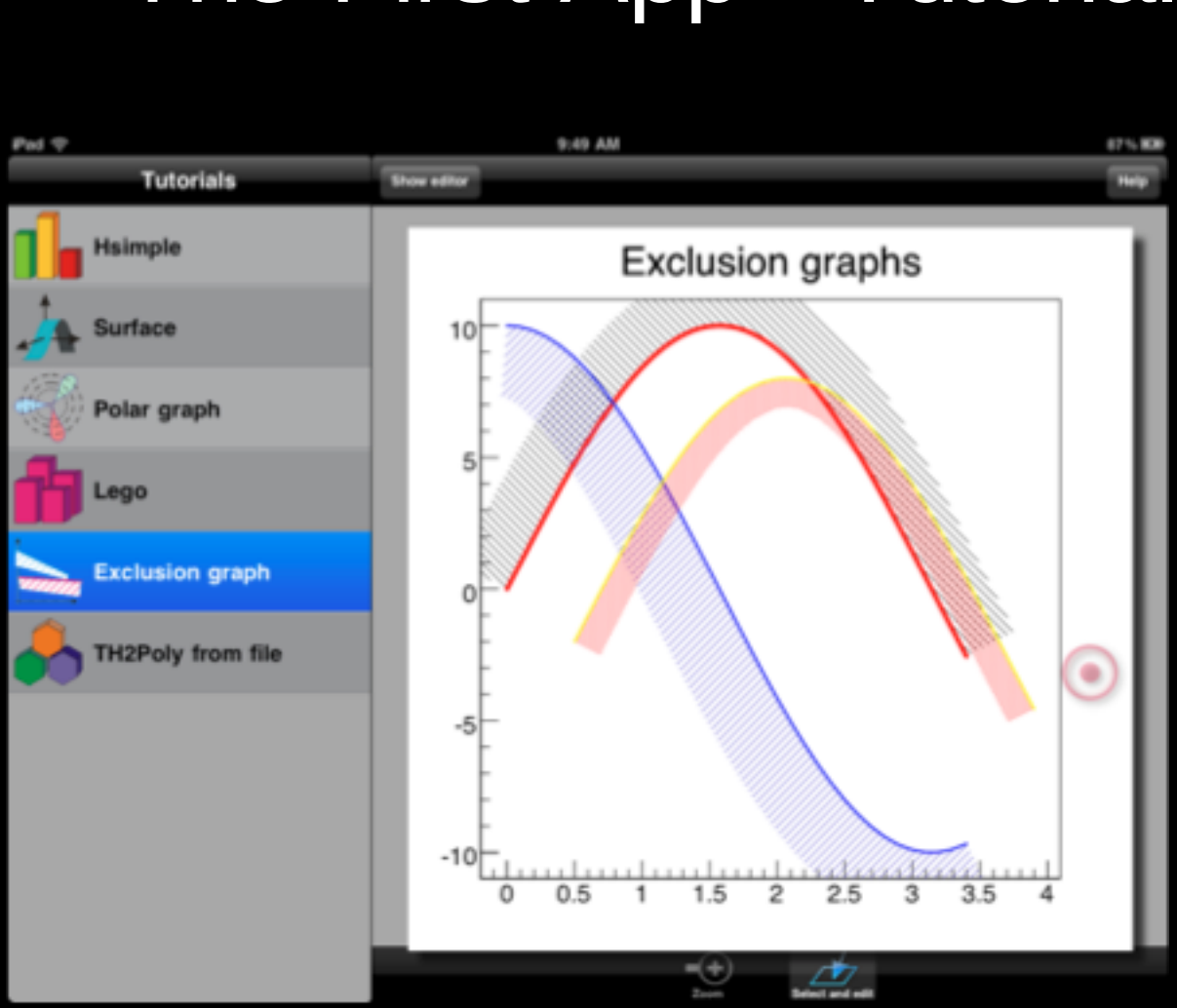


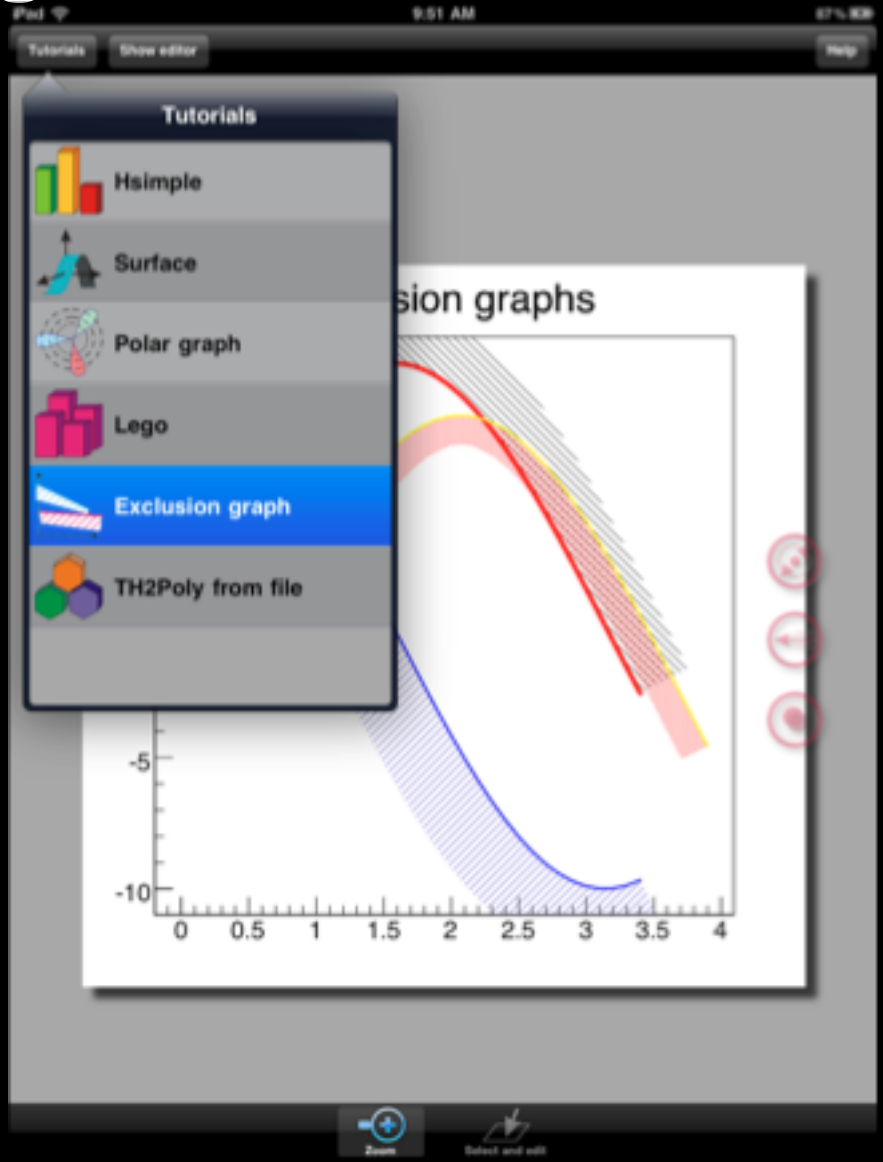


Using CoreText Framework

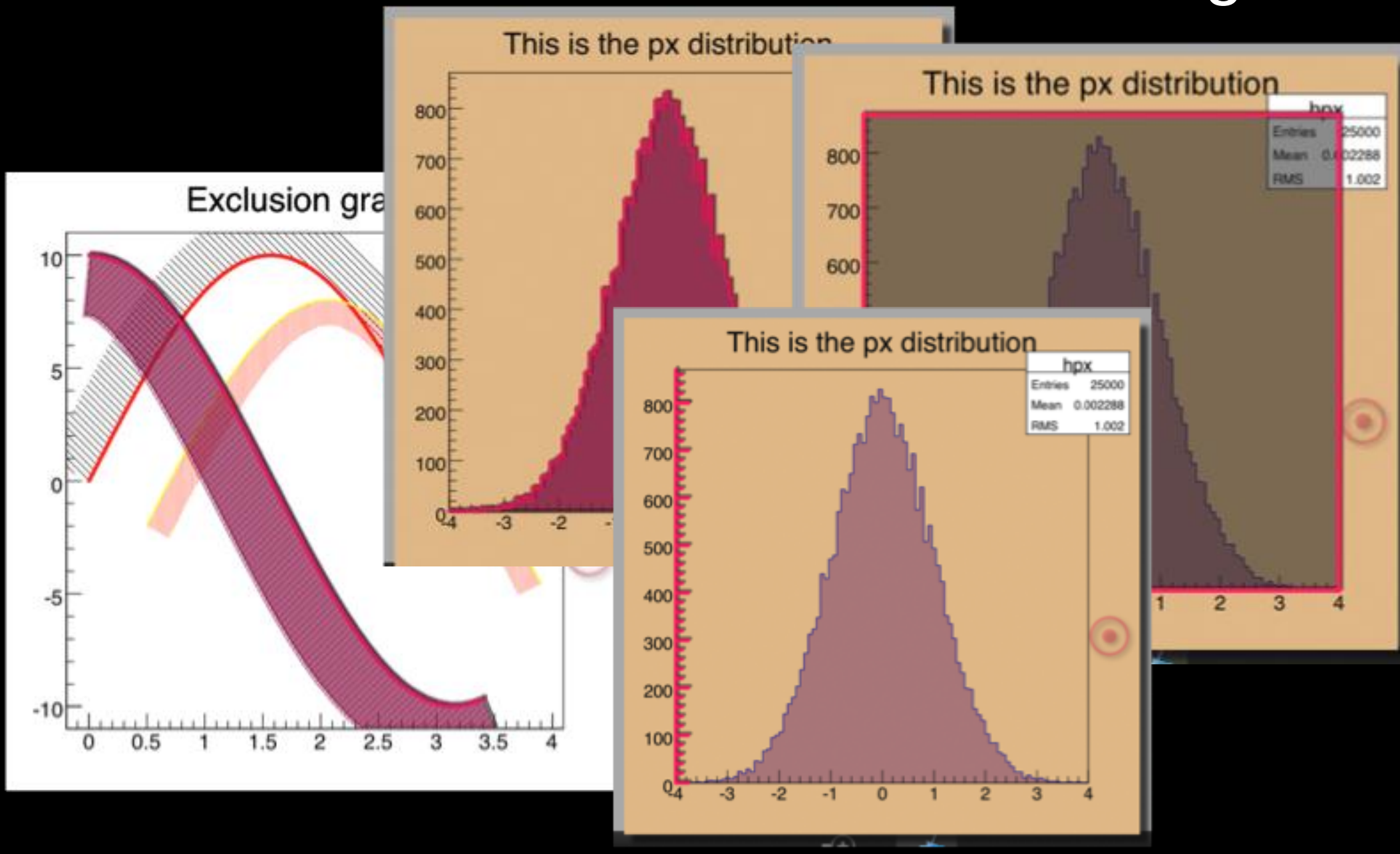


The First App - Tutorials

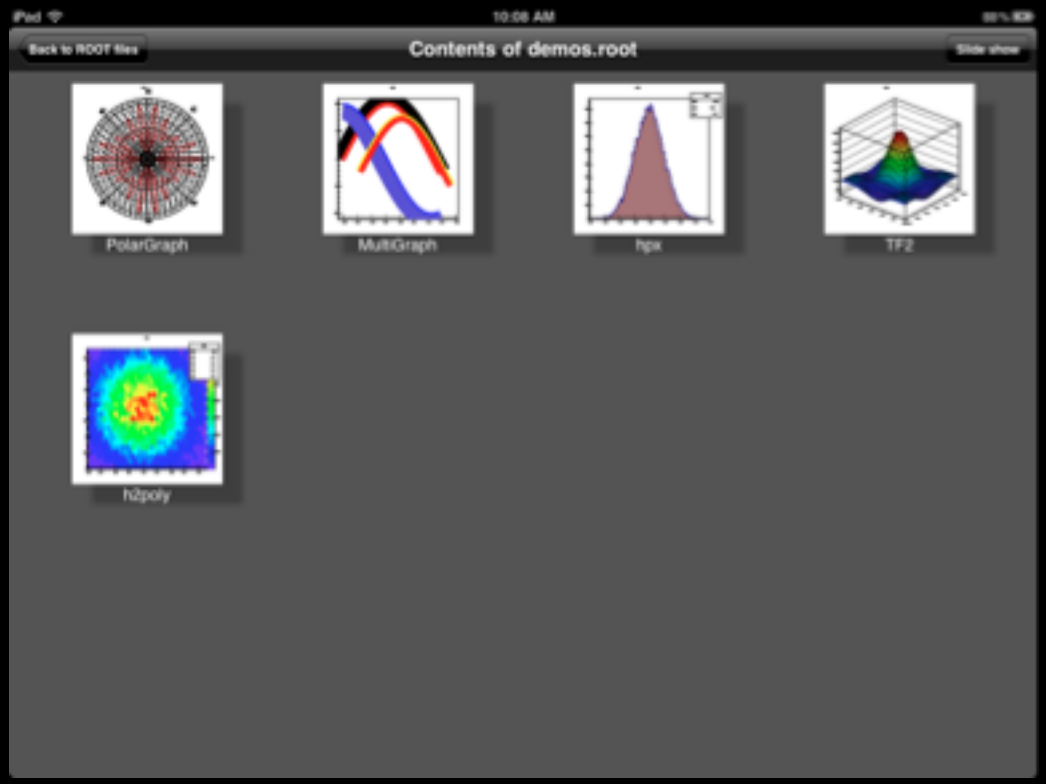


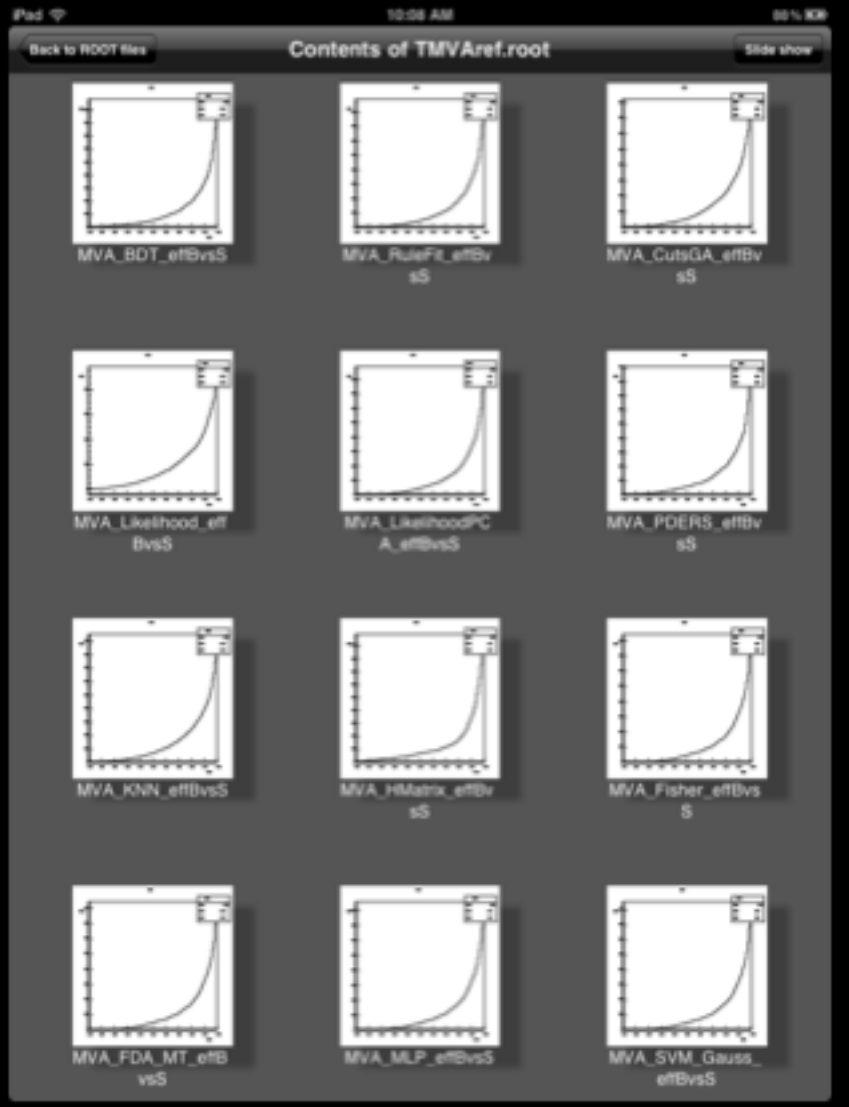


Extension to Pad to Visualize Picking

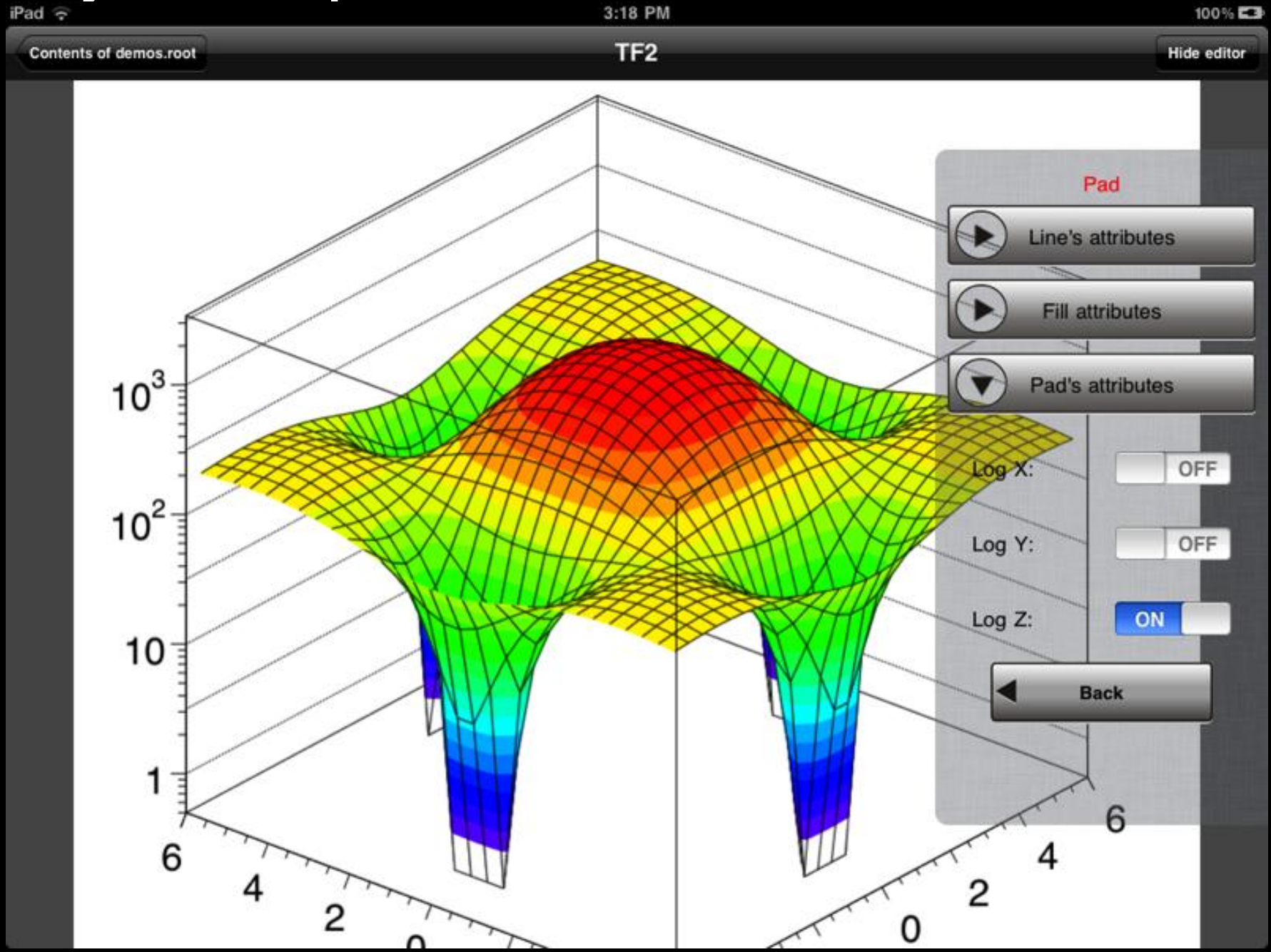


File Browser





Object Inspectors

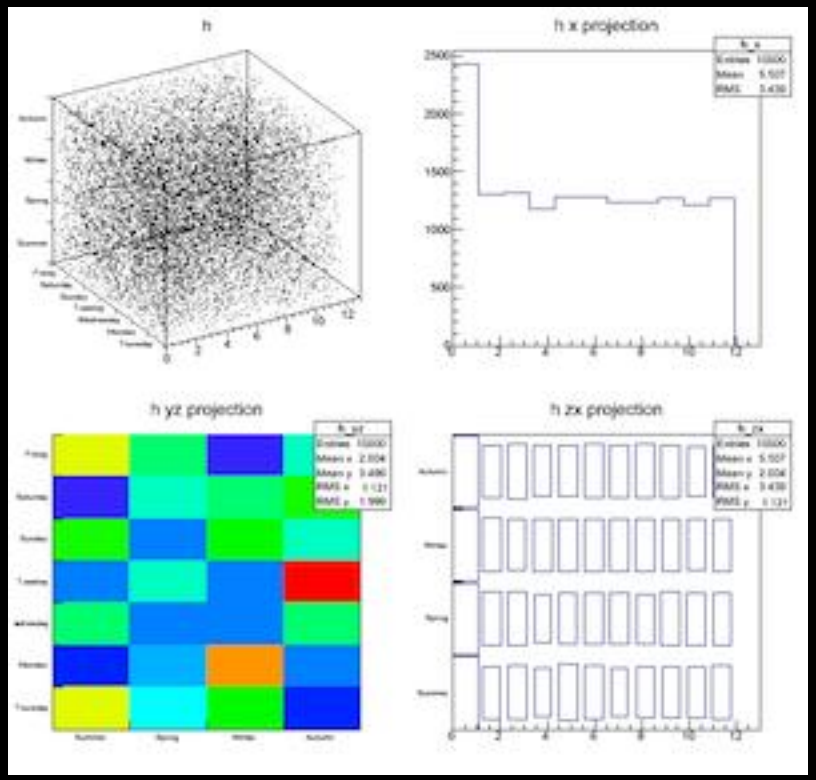


Browsers and HTML5

- Provide ROOT file access and graphics directly in the browser
  - ROOT-IO.js, ROOT-Graf.js, etc
- Avoids having to install ROOT as web server plugin (memory, security, etc. issues)
- Provide interactive graphics instead of static image display
- Would have also solved WebOS tablet support
  - But HP killed it
- Development beginning soon

Improvements in Graphics

- Move to native graphics on Mac OS X
- New more powerful TeX engine
- New refreshed look



New TeX Engine

New

$$\prod_{j\geq 0}\left(\sum_{k\geq 0} a_{jk}z^k\right) = \sum_{n\geq 0} z^n \left(\sum_{\substack{k_0,k_1,\ldots\geq 0 \\ k_0+k_1+\cdots=n}} a_{0k_0}a_{1k_1}\cdots \right.$$

$$W^{3\beta}_{\delta_1\rho_1\sigma_2} = U^{3\beta}_{\delta_1\rho_1\sigma_2} + \frac{1}{8\pi^2}\int_{a_1}^{a_2} da_2' \left[\frac{U^{2\beta}_{\delta_1\rho_1} - a_2' U^{1\beta}_{\rho_1\sigma_2}}{U^{0\beta}_{\rho_1\sigma_2}} \right]$$

$$= \frac{1}{2m_A}\left(\prod_f \frac{d^3p_f}{(2\pi)^3}\frac{1}{2E_f}\right)\left|\mathscr{M}\left(m_A - \{p_f\}\right)\right|^2 (2\pi)^4\delta^{(4)}\left(p_A - \sum\right.$$

$$4\mathrm{Re}\left\{\frac{2}{1-\Delta\alpha}\chi(s)\left[\hat{g}^e_v\hat{g}^f_v(1+\cos^2\theta) + \hat{g}^e_a\hat{g}^f_a\cos\theta\right]\right\}$$

$$p(n) = \frac{1}{\pi\sqrt{2}}\sum_{k=1}^{\infty}\sqrt{k}A_k(n)\frac{d}{dn}\frac{\sinh\left\{\frac{\pi}{k}\sqrt{\frac{2}{3}}\sqrt{n-\frac{1}{24}}\right\}}{\sqrt{n-\frac{1}{24}}}$$

RHIC スピン物理

Infrastructure

- gmake and now also cmake
- Drupal based website
- New documentation system
  - From MS Word to Docbook
- New source code convention checker
  - Eclair (commercial, but uses LLVM inside)
- New continuous build system
  - Electric Commander (commercial)
- Static code analyzer
  - Coverity (commercial)

ROOT Release v5-32-00

- Version v5-32-rc1 will be released Nov 1, 2011
- Version v5-32-rc2 will be released Nov 15, 2011
- Version v5-32-00 will be released Nov 29, 2011

Conclusions

- Users and experiments want an absolutely stable ROOT so their carefully tuned analysis are not disturbed when moving to a new version
- At the same time we need to extend to new platforms and incorporate and use new technologies to make life easier and “future proof” ROOT
- A careful balancing act